Dynamics of protein interaction subnetworks

Rubin, Katy Jane

Awarding institution:
King's College London

Thesis submitted to King's College London for the degree

of

Doctor of Philosophy

Dynamics of protein interaction subnetworks

Katy Jane Rubin

Abstract

I show that in the generic situations where a biological network, e.g. a protein interaction network, is in fact a subnetwork embedded in a larger "bulk" network, the presence of the bulk causes not just extrinsic noise but also memory effects. This means that the dynamics of the subnetwork will depend not only on its present state, but also its past. I use projection techniques to get explicit expressions for the memory functions that encode such memory effects, for generic protein interaction networks involving binary and unary reactions such as complex formation and phosphorylation, respectively. Remarkably, in the limit of low intrinsic copy-number noise such expressions can be obtained even for nonlinear dependences on the past. I illustrate the method with examples from a protein interaction network around epidermal growth factor receptor (EGFR), which is relevant to cancer signalling. These examples demonstrate that inclusion of memory terms is not only important conceptually but also leads to substantially higher quantitative accuracy in the predicted subnetwork dynamics.

I also study how the presence of Michaelis-Menten reactions affect the behaviour of the subnetwork. While such reactions do not directly fit into our framework of unary and binary reactions, I demonstrate that our approach can be generalised to include them. This is done by first introducing enzyme and enzyme complex species and reactions, then constructing the projected equations, and finally taking the limit of fast enzyme reactions that gives back Michaelis-Menten dynamics. I show that this limit can be taken in closed form, leading to a simple procedure that allows the projected equations to be constructed without ever introducing the fast variables explicitly.

I then apply projection methods to the analysis of the effects of perturbations in the bulk network, e.g. from gene regulation processes. I show that the resulting behaviour

of the linear response can again be decomposed according to a boundary structure, so that the total linear response is split into the effect of the bulk perturbation on the subnetwork boundary and the "propagation" of the perturbation from there to the rest of the subnetwork. I also use the projection method to find the steady states of the perturbed system in nonlinear response, which makes it possible to analyse biologically relevant scenarios such as knock-down experiments.

Finally, I look at the statistics of the random force. I propose a simple approximation of the random force made up of a persistent piece and a random change in the subnetwork initial conditions. I verify that this gives accurate predictions for both the linearised and nonlinear dynamics.

Acknowledgements

I would like to express my gratitude to the following people for their guidance over the past few years. Firstly I would like to thank my supervisor Peter Sollich for his help, patience and attention to detail. For their assistance and generosity I am grateful to Tony Ng, Franca Fraternali, Katherine Lawler and the rest of the Biologists who have provided the biological context for my work, as well as the Disordered Systems Group. Finally, I would like to thank my family and friends for their support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biological networks are often complex and models are required to try and understand their behaviour [1]. This has stimulated an ongoing research effort into the construction of reduced models that allow one to focus on subnetworks of a larger system. Such subnetworks may carry out biologically important functions, or be of interest because they capture parts of the system where there is less uncertainty in the network structure or dynamical parameters such as reaction rates. An understanding of the properties of such subnetworks can be used to help rationalise the behaviour of a larger network [2–4].

The above considerations motivate the analysis of subnetwork dynamics by model reduction, where one starts from a description of a large network and reduces this to an effective description of the subnetwork. Further motivation comes from the fact that almost any biological network that we choose to model is incomplete, and in reality is a subnetwork embedded in a larger "bulk" network. It is then important to understand what, in principle, is the appropriate way of describing the dynamics in such a subnetwork.

There is a substantial literature on methods of model reduction that attempt to simplify an initial large model down to a subnetwork description. The aim is to do this whilst retaining the main features of the behaviour of the original system [5, 6]. These methods are often based on (a) sensitivity analysis, (b) timescale separation, (c) splitting the system into modules or (d) lumping together components to obtain a smaller number of parameters or variables. A review of these methods is found in Chapter 2.

A further important topic is the effects of perturbations on a given subnetwork, e.g. via gene regulation. This is important both for a general understanding of biological function and more specifically to probe network structure and parameters like reaction rates [7, 8].

Alongside systematic effects from perturbations, the issue of fluctuations in biological networks has received much attention recently. Broadly these fluctuations arise from noise, which can be classified into intrinsic and extrinsic parts. In biochemical reaction networks, the intrinsic noise is caused by the stochasticity of when individual reaction events take place, causing fluctuations in the copy number of each molecular species. The extrinsic noise is due, for example, to fluctuations in reaction rates or amounts of other cellular components [9–11]. These noise components can be measured and separated using fluorescent proteins: for two identical copies of a gene, correlations in the fluctuations of the fluorescent proteins indicate extrinsic noise, while the intrinsic noise is obtained from the difference between the fluorescence of the two genes [12].

The main questions this thesis addresses are as follows. Can so-called projection methods be used to derive a systematic description of the dynamics of a subnetwork embedded in a "bulk" network? If this is feasible, does the conceptual division of a network into subnetwork and bulk also prove useful in understanding the response to perturbations? And can projection approaches provide information about the properties and statistics of extrinsic noise?

I find that the answer to all three questions is yes, though with some small qualifications. I construct explicit projected equations that describe the subnetwork; an important conceptual result is that such equations must in principle always involve memory effects, in addition to the extrinsic noise caused by the presence of the bulk. The memory function describing these effects can be expressed in closed form, though in a fully nonlinear description one requires the limit of small copy number (intrinsic) noise for this to be feasible. The projected equations also contain a contribution known as the random force that represents a component of the extrinsic noise, and because this is expressed explicitly in terms of (bulk) network parameters, the approach can be used to understand the statistics of this extrinsic noise component. In the analysis of perturbations I find that the projection approach allows for a natural separation of the response into two parts:

an initial propagation through the bulk arriving at the "boundary" of the subnetwork (those molecular species interacting directly with the bulk), followed by the response of the subnetwork to this effective boundary perturbation. This structure mirrors that of the memory effects, where only the dynamical equations of boundary species contain memory terms.

The structure of this thesis is as follows. In Chapter 2, I discuss the background material related to the later results. I will also discuss other methods of model reduction and subnetwork analysis. Then in Chapter 3, I explain the projection approach and how it can be applied to protein interaction networks. Remarkably, I obtain closed-form expressions of the memory functions for dynamics linearised around a fixed point in Sec. 3.2.1 and then for the full nonlinear dynamics in Sec. 3.2.2 when there is small copy number noise. In Sec. 3.2.3, I discuss and illustrate some of their properties, e.g. the amplitudes and timescales. I apply the method to the EGFR protein signalling network for short term signalling from Kholodenko et al. [13] and study the memory functions for a chosen subnetwork in Chapter 4. I analyse the dominant contributions to the memory functions and show that the projected equations with memory give a significantly more accurate description of the subnetwork dynamics than would be obtainable without memory.

The projection method works with mass action kinetics and to include Michaelis-Menten dynamics it is necessary to introduce additional species into the system to convert the reactions into mass action form. In Chapter 5, I discuss how we can extend the projection method so that it can be applied directly to reaction networks with Michaelis-Menten reactions without ever having to introduce additional species. I find that the closed form elimination is both faster to evaluate computationally, and gives a more accurate approximation to the original reaction equations.

I then extend the projection method further in Chapter 6 to include reactions that represent gene regulation. This allows us to analyse the effects of perturbations in the bulk network. I show that the resulting behaviour can again be decomposed according to a boundary structure, so that the total response is split into the effect of the bulk perturbation on the subnetwork boundary, and the "propagation" of the perturbation from there to the rest of the subnetwork.

The random force is a contribution to the projected equations that represents a component of extrinsic noise. In Chapter 7, I will discuss random force statistics. I will look at the accuracy of the projected equations when the bulk is not initially in steady state and how the random force can be approximated for both linearised and nonlinear dynamics.

Finally in Chapter 8, I will summarise the results I have presented and discuss potential improvements and extensions.

Chapter 2

Background

In this chapter I will discuss the methods used to describe biochemical networks. I will derive the projection method and review other methods of model reduction.

2.1 Reaction Equations

There are many different ways to model biochemical reactions [14–16]. Here we consider protein interaction networks where either two molecular species react to form a complex

$$i + j \to l \tag{2.1.1}$$

or change conformation from one state to another

$$i \to j \tag{2.1.2}$$

Complexes can also dissociate into two molecular species

$$l \to i + j \tag{2.1.3}$$

Let n_i be the number of molecules of species i. Then we can write the chemical master equation [17, 18] as

$$\frac{\partial}{\partial t}P(\boldsymbol{n},t) = V\sum_{\mu=1}^{R}\left(\prod_{i=1}^{N}E_i^{-S_{i\mu}} - 1\right)\tilde{f}_\mu(\boldsymbol{n},V)P(\boldsymbol{n},t) \tag{2.1.4}$$

where E_i is a step operator defined such that $E_i^{-1} = n_i - 1$. $\boldsymbol{S}$ is the stoichiometry matrix made up of integers $S_{i\mu}$, with $i = 1, 2, \ldots, N$ and $\mu = 1, 2, \ldots, R$, where R is the number

of reactions. Each $S_{i\mu}$ records by how much the molecule count of species i changes in reaction μ. Specifically, $S_{i\mu}$ is -1 if species i is a reactant in reaction μ and $+1$ if it is a reaction product. For homodimer reactions one correspondingly has $S_{i\mu} = \pm 2$ when two molecules of species i are used up or produced. Let k_μ be the reaction rate for a reaction R_μ then $\tilde{f}_\mu(\boldsymbol{n}, V)$ denotes the reaction fluxes which are $k_\mu n_i/V$ for a unary reaction, $k_\mu n_i n_j/V^2$ for a binary reaction between species i and j and $(1/2)k_\mu n_i(n_i-1)/V^2$ for a binary reaction between two of the same species i.

In the limit when the volume V is large enough so that stochastic calculations around the mean value can be neglected we can write a set of deterministic reaction equations. If the concentration of species i is $x_i = n_i/V$ we can then write the reaction equations in terms of the stoichiometry matrix and reaction flux vector $\boldsymbol{f}$ as

$$\frac{\partial}{\partial t} x_i = \sum_\mu S_{i\mu} f_\mu(\boldsymbol{x}) \tag{2.1.5}$$

The vector of reaction fluxes $\boldsymbol{f}$ has entries f_μ that give the reaction rate of reaction μ multiplied by the concentrations of the proteins involved in that reaction.

The chemical master equation (2.1.4) is hard to analyse and therefore approximations to the master equation are needed. The chemical Fokker-Planck equation is obtained by truncating the Kramers-Moyal expansion of the chemical master equation [19] to include at most second-order derivatives. Taylor expanding the step operator we find

$$\begin{aligned}
\frac{\partial}{\partial t} P(\boldsymbol{n}, t) &= V \sum_{\mu=1}^{R} \left(\prod_{i=1}^{N} E_i^{-S_{i\mu}} - 1 \right) \tilde{f}_\mu(\boldsymbol{n}, V) P(\boldsymbol{n}, t) \\
&= V \sum_{\mu=1}^{R} \left(\prod_{i=1}^{N} e^{-S_{i\mu}\partial/\partial n_i} - 1 \right) \tilde{f}_\mu(\boldsymbol{n}, V) P(\boldsymbol{n}, t) \\
&\simeq V \sum_{\mu=1}^{R} \left(-\sum_{i=1}^{N} S_{i\mu} \frac{\partial}{\partial n_i} + \frac{1}{2} \sum_{i,j=1}^{N} S_{i\mu} S_{j\mu} \frac{\partial^2}{\partial n_i \partial n_j} \right) \tilde{f}(\boldsymbol{n}, V) P(\boldsymbol{n}, t)
\end{aligned}$$

Transforming the variables from number of molecules into concentrations such that $\boldsymbol{x} = \boldsymbol{n}/V$ this becomes

$$\frac{\partial}{\partial t} P(\boldsymbol{x}, t) = \sum_{\mu=1}^{R} \left(-\sum_{i=1}^{N} S_{i\mu} \frac{\partial}{\partial x_i} + \frac{1}{2V} \sum_{i,j=1}^{N} S_{i\mu} S_{j\mu} \frac{\partial^2}{\partial x_i \partial x_j} \right) f(\boldsymbol{x}) P(\boldsymbol{x}, t)$$

and we can write the Fokker-Planck equation as

$$\frac{\partial}{\partial t} P(\boldsymbol{x}, t) = -\frac{\partial}{\partial \boldsymbol{x}} (\boldsymbol{S}\boldsymbol{f} P(\boldsymbol{x}, t)) + \epsilon \frac{\partial^2}{\partial \boldsymbol{x}^2} (\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} P(\boldsymbol{x}, t)) = \mathcal{L}^{\mathrm{T}} P(\boldsymbol{x}, t) \tag{2.1.6}$$

where $\epsilon = 1/V$ is the inverse reaction volume. We define $\boldsymbol{Sf}$ as the drift vector and

$$\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} = \boldsymbol{S}\mathrm{diag}(\boldsymbol{f})\boldsymbol{S}^{\mathrm{T}} \tag{2.1.7}$$

is the diffusion matrix. This formulation is useful for us as we can continue to describe each species concentration with a single variable x_i, rather than having to treat its mean time evolution and fluctuations separately as would be done in a van Kampen system size expansion [17, 20]. Moreover, a recent analysis [21] shows that (2.1.6) is more accurate than the van Kampen description, capturing the mean and variance of the x_i to higher order in ϵ.

We will sometimes find it useful to switch from the above Fokker-Planck description to the corresponding "chemical Langevin equation" [18], which reads

$$\frac{\partial}{\partial t}\boldsymbol{x} = \boldsymbol{S}\boldsymbol{f}\boldsymbol{x} + \boldsymbol{\eta} \tag{2.1.8}$$

The noise $\boldsymbol{\eta}$ is multiplicative as its statistics depend on $\boldsymbol{x}$; adopting the Ito interpretation [19], one has explicitly $\langle \boldsymbol{\eta}(t)\boldsymbol{\eta}^{\mathrm{T}}(t')\rangle = \epsilon \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\delta(t-t')$.

It is also possible to model chemical reactions using a path integral representation of which there are a number of different methods [22–25]. This method uses the protein concentrations $\boldsymbol{x}$ as well as a set of auxiliary variables $\hat{\boldsymbol{x}}$. Discretising time so that Δ represents a time step, we then define a generating functional [26]

$$Z[\boldsymbol{\psi}] = \int \prod_{it} \frac{d\eta_i(t)dx_i(t)d\hat{x}_i(t)}{2\pi} e^{i\Delta\psi_i(t)x_i(t)} \cdot e^{i\hat{x}_i(t)[x_i(t+\Delta)-x_i(t)-\Delta\Phi_i(\boldsymbol{x}(t))-\Delta\eta_i(t)]} P(\boldsymbol{\eta}(t)) \tag{2.1.9}$$

where $\Phi_i(\boldsymbol{x}(t)) = \boldsymbol{S}\boldsymbol{f}\boldsymbol{x}$. Then splitting both the $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ into subnetwork and bulk species we can integrate out the bulk variables which gives an effective path integral describing the subnetwork dynamics. This also allows us to explicitly keep the intrinsic noise when calculating the subnetwork equations.

2.2 The projection method

The Zwanzig-Mori projection method is used to describe the time evolution of a set of observables from a chosen subnetwork embedded in a larger network [27–29]. Generally mode-coupling approximations are applied to the projected equations [30]. This simplifies the projected equations allowing them to be analysed easily. Mode coupling theory

is used to predict the features of the dynamics of liquids near the glass transition. Here we use the projected equations to describe the dynamics of protein interaction networks.

Returning to the Fokker-Planck equation (3.1.2), the time evolution it encodes can be thought of in terms of either an evolving $P(\boldsymbol{x},t)$ or evolving observables $a(\boldsymbol{x},t)$ of the system; see e.g.[27, 29]. The time variation of $P(\boldsymbol{x},t)$ is the solution of (3.1.2), which can be written formally as $P(\boldsymbol{x},t) = e^{\mathcal{L}^{\mathrm{T}}t}P(\boldsymbol{x},0)$. Here the operator exponential in $e^{\mathcal{L}^{\mathrm{T}}t}$ is defined as $e^{\mathcal{L}^{\mathrm{T}}t} = \sum_{n=0}^{\infty}(\mathcal{L}^{\mathrm{T}}t)^n/n!$, requiring in principle the application of successive powers of $\mathcal{L}^{\mathrm{T}}t$ to $P(\boldsymbol{x},0)$.

Now let $a(\boldsymbol{x})$ be an observable of the system, for example one of the protein concentrations x_i. Its time average evolves in time as

$$\begin{aligned}\langle a(t)\rangle &= \int d\boldsymbol{x}\; a(\boldsymbol{x})P(\boldsymbol{x},t) = \int d\boldsymbol{x}\; a(\boldsymbol{x})e^{\mathcal{L}^{\mathrm{T}}t}P(\boldsymbol{x},0)\\ &= \int d\boldsymbol{x}\; e^{\mathcal{L}t}a(\boldsymbol{x})P(\boldsymbol{x},0) = \int d\boldsymbol{x}\; a(\boldsymbol{x},t)P(\boldsymbol{x},0)\end{aligned} \tag{2.2.1}$$

Here we have introduced $\mathcal{L}$ as the adjoint operator to $\mathcal{L}^{\mathrm{T}}$, defined by $\int d\boldsymbol{x}\,(\mathcal{L}a(\boldsymbol{x}))b(\boldsymbol{x}) = \int d\boldsymbol{x}\, a(\boldsymbol{x})\mathcal{L}^{\mathrm{T}}b(\boldsymbol{x})$. We have also defined

$$a(\boldsymbol{x},t) = e^{\mathcal{L}t}a(\boldsymbol{x}) \tag{2.2.2}$$

As the last equality of (2.2.1) shows, this is the average value of a at time t conditional on the system initially being in state $\boldsymbol{x}$. Its time evolution is given by (2.2.2), and reads in differential form

$$\partial_t a(\boldsymbol{x},t) = \mathcal{L}a(\boldsymbol{x},t) \tag{2.2.3}$$

with initial condition $a(\boldsymbol{x},0) = a(\boldsymbol{x})$.

The Zwanzig-Mori projection method allows us to derive a set of equations describing the time evolution of observables from a chosen subnetwork [27–29]. The approach allows one generally to derive such equations for the conditional averages $a_\alpha(\boldsymbol{x},t)$ of any chosen set of observables $\{a_\alpha(\boldsymbol{x})\}$. One first defines a projection operator $\mathcal{P}$ that projects any observable b onto the space spanned by the chosen set of observables:

$$(\mathcal{P}b)(\boldsymbol{x}) = \sum_{\alpha,\beta} a_\beta(\boldsymbol{x})\left(\boldsymbol{C}^{-1}\right)_{\beta\alpha}(a_\alpha, b) \tag{2.2.4}$$

Here $\boldsymbol{C}$ is a correlation matrix with elements

$$C_{\alpha\beta} = (a_\alpha, a_\beta) \tag{2.2.5}$$

defined in terms of an inner product (a, b). The latter is just an average over the steady state distribution $P_{\rm ss}(\boldsymbol{x})$ of $\boldsymbol{x}$:

$$(a, b) = \langle ab \rangle_{\rm ss} = \int d\boldsymbol{x}\ a(\boldsymbol{x}) b(\boldsymbol{x}) P_{\rm ss}(\boldsymbol{x}). \tag{2.2.6}$$

Once $\mathcal{P}$ is defined, the orthogonal projection operator $\mathcal{Q}$ follows as $\mathcal{Q} = 1 - \mathcal{P}$. Then $\mathcal{Q}b$ can be interpreted as the contribution to observable b that is uncorrelated in steady state with any of the chosen observables a_α.

To derive the projected equations we start be taking the Laplace Transform of equation (2.2.3). Writing $(z - \mathcal{L}) = (z - \mathcal{P}\mathcal{L} - \mathcal{Q}\mathcal{L})$ we find

$$\begin{aligned} z a_\alpha(z) - a_\alpha &= (z - \mathcal{P}\mathcal{L} - \mathcal{Q}\mathcal{L})^{-1} \mathcal{L} a_\alpha \\ &= (z - \mathcal{L})^{-1} \mathcal{P}\mathcal{L} a_\alpha + (z - \mathcal{L})^{-1} \mathcal{P}\mathcal{L} (z - \mathcal{Q}\mathcal{L})^{-1} \mathcal{Q}\mathcal{L} a_\alpha + (z - \mathcal{Q}\mathcal{L})^{-1} \mathcal{Q}\mathcal{L} a_\alpha \\ &= \sum_\beta a_\beta(z) \Omega_{\beta\alpha} + \sum_\beta a_\beta(z) M_{\beta\alpha}(z) + (z - \mathcal{Q}\mathcal{L})^{-1} \mathcal{Q}\mathcal{L} a_\alpha \end{aligned}$$

where we have used the matrix equality

$$(A - B)^{-1} = (A - B)^{-1} B A^{-1} + A^{-1} \tag{2.2.7}$$

Transforming back into the time domain we obtain the projected equations

$$\frac{\partial}{\partial t} a_\alpha(t) = \sum_\beta a_\beta(t) \Omega_{\beta\alpha} + \int_0^t dt' \sum_\beta a_\beta(t') M_{\beta\alpha}(t - t') + r_\alpha(t) \tag{2.2.8}$$

The first term on the r.h.s. is local in time. We will call the coefficients

$$\Omega_{\beta\alpha} = \sum_\gamma \left(\boldsymbol{C}^{-1}\right)_{\beta\gamma} (a_\gamma, \mathcal{L} a_\alpha) \tag{2.2.9}$$

the elements of the rate matrix $\boldsymbol{\Omega}$; in other contexts, e.g. systems with inertial dynamics, it is often referred to as the frequency matrix. The second term represents the memory effects, as an integral over past values of the observables weighted by a function of the time lag, the *memory function*. The latter can be expressed as

$$M_{\beta\alpha}(\Delta t) = \sum_\gamma \left(\boldsymbol{C}^{-1}\right)_{\beta\gamma} \left(a_\gamma, \mathcal{L}\mathcal{Q} e^{\mathcal{Q}\mathcal{L}\mathcal{Q}\Delta t} \mathcal{Q}\mathcal{L} a_\alpha\right) \tag{2.2.10}$$

where $\Delta t = t - t'$. The memory function $M_{\beta\alpha}$ determines how strongly the past values of observable a_β affect the present rate of change of a_α; sometimes it will be useful to think of the $M_{\beta\alpha}(\Delta t)$ as the elements of a memory matrix $\boldsymbol{M}(\Delta t)$ whose size is, as

for the rate matrix, the number of observables a_α. The third term in (2.2.8), finally, is called the random force and is written

$$r_\alpha(t) = e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t}\mathcal{Q}\mathcal{L}a_\alpha. \tag{2.2.11}$$

The name comes from the fact that the value of $r_\alpha(t)$ at any time t is uncorrelated with the initial values of the observables $a_\beta(0) \equiv a_\beta$; mathematically this property is expressed as $(a_\alpha, r_\beta(t)) = 0$.

2.3 Subnetworks

In this thesis we use the projection method to obtain a set of equations for a subnetwork embedded in a larger protein interaction network. Below we review other methods of model reduction that attempt to simplify an initial large model down to a subnetwork description.

Sensitivity analysis tries to determine which molecular species are insignificant to the dynamic system of interest [31]. A parameter is classified as insignificant if it has a low sensitivity, in that its precise value does not have a large effect on the concentrations of the rest of the species in the network. If we consider a reaction system

$$\frac{\partial}{\partial t}\boldsymbol{x} = \boldsymbol{S}\boldsymbol{f}\boldsymbol{x} = g(\boldsymbol{x}) \tag{2.3.1}$$

then we can determine the importance of species i by considering

$$B_i = \sum_j \left(\frac{\partial \ln g_j}{\partial \ln x_i}\right)^2 \tag{2.3.2}$$

where j indexes the components of $\boldsymbol{x}$. Then any species i with sensitivity $B_i < B$, where B is a parameter reflecting the desired accuracy of the model, can be considered to have a low sensitivity. Low sensitivity parameters are then eliminated or replaced by a smaller number of effective species. However, sometimes it is necessary to keep a low sensitivity parameter to ensure the results are biologically valid. This approach will not work with a predetermined subnetwork as that subnetwork may contain a mixture of low and high sensitivity parameters and therefore the subnetwork found using sensitivity analysis is effectively chosen by the method.

Timescale separation techniques are used to focus on the species that contribute most to the long-time dynamics of a system, by removing molecular species whose dynamics takes place on much shorter timescales. This is reasonable because biochemical processes occur on a range of timescales; changes in gene expression levels, for example, may take place over hours whereas protein signalling takes seconds. Timescale separation approaches have been used by e.g. Gardiner [32] and Thomas et al. [33], with the subnetwork then containing all the slow molecular species and the bulk the fast ones. Thus, while these authors used projection techniques as we do, memory effects did not arise: they become negligible if the bulk is fast enough to respond effectively instantaneously – on the timescale of the subnetwork dynamics – to the state of the subnetwork. Here we consider signalling networks where the timescales of the dynamics of the subnetwork and the bulk are comparable, so that timescale separation methods are not directly applicable.

Another way to reduce the system is to split it into modules where each module has a different function and a limited number of interactions with the other modules [34]. Conzelmann et al. [35] split a network into modules with low retroactivity, where two modules are connected without retroactive effects if both the input and output are unidirectional [36]. These modules can then be uncoupled from the larger network as their behaviour only depends on inputs. They then apply dimensional reduction to the modules so that the nonlinear modules have reduced complexity but show similar input and output behaviour. To apply this method it is necessary to have a detailed knowledge of the system dynamics and one must be able to decompose the model into modules with low retroactivity. In general it may be possible to fix a subnetwork and split the subnetwork and bulk into modules; however if, for example, there is a subnetwork with many species directly interacting with the bulk, then it may not be easy to separate the fixed subnetwork into modules containing no bulk species.

Lumping together variables with similar features also allows one to reduce the size of a model [5, 37]; however, lumping components together may make it difficult to interpret the results because the lumped variables may not retain their original meaning. Sunnaker et al. [37] split their system dependent on whether the variables are significant for the fast or slow dynamics. The lumped state variables, which are the variables of the reduced model, correspond to clusters of fast variables that are associated to conservation laws.

They then derive fraction parameters which link the reduced and original model. The reduced model is used to analyse the dynamics of the system and then the lumped variables are translated back into the original variables. Similarly Liebermeister et al. [38] reduce the bulk surrounding a chosen subnetwork, whilst the subnetwork is kept in its original form. They first linearise the bulk around a steady state and then apply balanced truncation to reduce the dimensionality of the linearised bulk. As one might expect, accounting for the bulk in this way, i.e. considering the environment surrounding the subnetwork, yields a reduced model that is more accurate than modelling just the isolated subnetwork. Our work extends this result by showing that the inclusion of memory effects arising from the bulk gives a significantly more accurate description of the subnetwork dynamics. Apri et al. [39] remove or modify reactions and parameters based on their effect on the output behaviour of the system. They consider which parameters can be removed or lumped together to obtain output data correct to within a certain tolerance. Although no detailed prior biological knowledge of the system is needed, there must be some qualitative understanding of the system dynamics to ensure no species which are generally considered to be an important part of the network dynamics are removed.

Chapter 3

Projection method

I now explain how the projection method can be applied to protein interaction networks. I also illustrate the method with a simple example that already captures some general properties of memory functions. Next, in Section 3.2, I derive the closed-form memory functions for both the linearised dynamics and the full nonlinear dynamics. I then discuss and illustrate some of their properties, e.g. the amplitudes, and what they tell us about reactions between the subnetwork and the bulk.

3.1 Derivation of projected equations

3.1.1 Reaction equations

We consider a protein interaction network described using mass action kinetics. The molecular reactions can be either binary or unary. In a binary reaction two molecules react to form a molecule of a different species (complex formation); the reverse process is the dissociation of a complex into two molecules. In a unary reaction, one species transforms into another via a conformational change like phosphorylation. In our setup we do not restrict the nature of the molecules that come together in a binary reaction, and in particular we include the possibility that a complex formed in some initial binary reaction may react again with another molecule to form a higher order complex. As a convenient notational shorthand we nevertheless refer generically to the two molecules that join together in a binary reaction as "proteins", and to the molecule that is formed

as a complex.

The deterministic reaction equations for such a protein interaction network containing N molecular species can be written in the form

$$\begin{aligned}\frac{\partial}{\partial t}x_i = &\sum_{j(\neq i),l}\left(k^-_{l,ij}x_l - k^+_{ij,l}x_ix_j\right) + \frac{1}{2}\sum_{j\neq l}\left(k^+_{jl,i}x_jx_l - k^-_{i,jl}x_i\right)\\ &+\sum_l\left(2k^-_{l,ii}x_l - k^+_{ii,l}x_ix_i\right) + \sum_j\left(\frac{1}{2}k^+_{jj,i}x_jx_j - k^-_{i,jj}x_i\right)\\ &+\sum_j\left(\lambda_{ji}x_j - \lambda_{ij}x_i\right)\end{aligned} \tag{3.1.1}$$

where x_i is the concentration of species i. In our notation we follow to a large extent the paper by Coolen and Rabello [40], which presented an average-case analysis using generating functionals of the dynamics in large protein interaction networks. We denote by $k^+_{ij,l}$ the rate of formation of complex l from proteins i and j, and by $k^-_{l,ij}$ the rate for the reverse process of dissociation of complex l into proteins i and j. To avoid ordering restrictions on the protein indices we set $k^+_{ij,l} = k^+_{ji,l}$ and $k^-_{l,ij} = k^-_{l,ji}$. The factor of $1/2$ in the first line above is then needed to avoid double counting of reactions of two different molecular species. The second line describes homodimer formation and dissociation, where two proteins of the same species i react. The extra factor of 2 arises because dissociation of a homodimer l creates two molecules of species i. The factor $1/2$ in the term describing formation of i from two molecules of species j represents the reduction in number of possible reaction pairs, compared to the case of formation of a heterodimer where the two reacting species are different. The unit prefactor of the $k^+_{ii,l}$ term arises as the combination of these two effects. Finally, the last line of (3.1.1) accounts for unary reactions, with λ_{ij} the rate of species i changing into species j.

The reaction equations (3.1.1) can be written in terms of a stoichiometry matrix $\boldsymbol{S}$ and vector of reaction fluxes $\boldsymbol{f}$ in the compact form $\partial_t x_i = \sum_\mu S_{i\mu}f_\mu$ as shown in Section 2.1. One benefit of this formulation is that it shows transparently how conservation laws arise, where the sum of a number of concentrations is constant in time. Quantitatively, the number of conservation laws is given by the dimension of the left nullspace of $\boldsymbol{S}$. If this nullspace is spanned by the (column) vectors $\boldsymbol{e}^{(a)}$, $a = 1, 2, \ldots$ then each such vector obeys $\boldsymbol{e}^{(a)\mathrm{T}}\boldsymbol{S} = 0$. Accordingly the quantity $\sum_i x_i e^{(a)}_i$ is conserved: $\partial_t \sum_i x_i e^{(a)}_i = \boldsymbol{e}^{(a)\mathrm{T}}\boldsymbol{S}\boldsymbol{f} = 0$.

3.1.2 Stochastic Dynamics

The deterministic reaction equations (3.1.1) apply in the case where the number of molecules of each species, $x_i V$ in a reaction compartment of volume V, is large enough so that stochastic fluctuations around the mean value can be neglected. In reality such copy number fluctuations are always present because the number of molecules of any species is discrete, and when it changes over time it does so due to elementary reactions that take place stochastically. The relative size of the fluctuations in any x_i will be of order $1/\sqrt{x_i V}$, because any change in x_i results from the cumulative effect of many reactions and the number of reactions occurring within any fixed time interval grows linearly with V.

We therefore next describe the stochastic extension of (3.1.1) to the case of small copy number fluctuations. The inverse volume of the system, $\epsilon = 1/V$, will be used to characterize the strength of this intrinsic noise. We note that such a stochastic description is also important for our use of the projection operator formalism [27] to derive subnetwork dynamical equations, as this approach starts from the time evolution of a probability distribution over states of the network.

For small ϵ, the appropriate stochastic version of (3.1.1) is a Fokker-Planck equation for the time evolution of the probability density $P(\boldsymbol{x}, t)$ as derived in Sec. 2.1 where truncating a Kramers-Moyal expansion [19] after the first order in ϵ we obtained

$$\frac{\partial P(\boldsymbol{x}, t)}{\partial t} = -\frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{S}\boldsymbol{f} P\right) + \frac{\epsilon}{2} \frac{\partial^2}{\partial \boldsymbol{x}^2} \left(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} P\right) = \mathcal{L}^{\mathrm{T}} P(\boldsymbol{x}, t) \tag{3.1.2}$$

In Section 2.2 we then showed that the time evolution of an observable $a(\boldsymbol{x})$ is given by (2.2.2) and reads in differential form

$$\partial_t a(\boldsymbol{x}, t) = \mathcal{L} a(\boldsymbol{x}, t) \tag{3.1.3}$$

and therefore to calculate the projected equations we need to write down the adjoint Fokker-Planck operator.

Before we write down the adjoint Fokker-Planck operator, we make a change of variables. For reasons explained further in Section 3.1.3 below, it will be useful to have variables with a mean value of zero in steady state. We therefore define $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{\delta x}$ where $\boldsymbol{y}$ is the mean steady state value of $\boldsymbol{x}$, calculated as the fixed point of the mass-action equations

(3.1.1), and $\boldsymbol{\delta x}$ is the deviation away from this. Where the meaning is clear from the context, we will often use the shorthand "concentration" for the concentration deviations from steady state, δx_i. We have defined $\boldsymbol{y}$ to be the mean steady state value of $\boldsymbol{x}$ and we therefore require our system to be monostable. Furthermore, $\boldsymbol{y}$ is the fixed point of the rate equations and therefore $\langle \boldsymbol{x} \rangle = 0$ is only true in the limit of small intrinsic noise. The time-evolving probability distribution is now $P(\boldsymbol{\delta x}, t)$, and observables $a(\boldsymbol{\delta x})$ are likewise functions of $\boldsymbol{\delta x}$. In terms of these variables the adjoint Fokker-Planck operator $\mathcal{L}$ writes

$$\begin{aligned}
\mathcal{L} = & \sum_{i,j,l:i\neq j} \left[k^-_{l,ij} \delta x_l - k^+_{ij,l} (y_j \delta x_i + y_i \delta x_j + \delta x_i \delta x_j) \right] \frac{\partial}{\partial \delta x_i} \\
& + \frac{1}{2} \sum_{i,j,l:j\neq l} \left[k^+_{jl,i} (y_l \delta x_j + y_j \delta x_l + \delta x_j \delta x_l) - k^-_{i,jl} \delta x_i \right] \frac{\partial}{\partial \delta x_i} \\
& + \sum_{i,j} \Bigg\{ \left[2k^-_{j,ii} \delta x_j - k^+_{ii,j} (2y_i \delta x_i + \delta x_i \delta x_i) \right] \\
& + \frac{1}{2} \left[k^+_{jj,i} (2y_j \delta x_j + \delta x_j \delta x_j) - k^-_{i,jj} \delta x_i \right] \Bigg\} \frac{\partial}{\partial \delta x_i} \\
& + \sum_{i,j} (\lambda_{ji} \delta x_j - \lambda_{ij} \delta x_i) \frac{\partial}{\partial \delta x_i} + \frac{\epsilon}{2} \sum_{i,j} \left(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} \right)_{ij} \frac{\partial^2}{\partial \delta x_i \, \partial \delta x_j}
\end{aligned} \tag{3.1.4}$$

All terms here except for the last describe deterministic evolution. To write the reaction flux prefactors from (3.1.1) we have replaced $x_i = y_i + \delta x_i$ and exploited the fact that when $\boldsymbol{x} = \boldsymbol{y}$, i.e. $\boldsymbol{\delta x} = 0$, the deterministic drift terms must vanish. Note that $\mathcal{L}c = 0$ for any constant c, so that from (2.2.3) the average of such an "observable" is constant in time as it should be. Looking at (2.2.1), this property is equivalent to conservation of probability in the original Fokker-Planck equation.

3.1.3 Projection

We next summarise the salient features of the Zwanzig-Mori projection method we use to derive equations describing the time evolution of the concentrations in any chosen subnetwork of a larger protein interaction network [27–29]. For the full derivation see Sec. 2.2. In this approach we obtain a set of equations for the conditional averages $a_\alpha(\boldsymbol{\delta x}, t)$ of any chosen set of observables $\{a_\alpha(\boldsymbol{\delta x})\}$. In our case the $\{a_\alpha(\boldsymbol{\delta x})\}$ will be a set of observables from the system such as protein and complex concentrations from the

subnetwork, as discussed in more detail below. First a projection operator $\mathcal{P}$ is defined

$$(\mathcal{P}b)(\boldsymbol{\delta x}) = \sum_{\alpha,\beta} a_\beta(\boldsymbol{\delta x}) \left(\boldsymbol{C}^{-1}\right)_{\beta\alpha} (a_\alpha, b) \tag{3.1.5}$$

This projects any observable b onto the space spanned by the chosen set of observables, where the inner product (a, b) is an average over the steady state distribution of $\boldsymbol{\delta x}$:

$$(a, b) = \langle ab \rangle_{\text{ss}} = \int d\boldsymbol{\delta x}\ a(\boldsymbol{\delta x}) b(\boldsymbol{\delta x}) P_{\text{ss}}(\boldsymbol{\delta x}). \tag{3.1.6}$$

We see now explicitly that we need stochastic dynamics, i.e. nonzero ϵ, to be able to deploy the projection formalism, even if we are interested in the limit of small ϵ. If we were to set $\epsilon = 0$ directly, the steady state distribution would become a Dirac delta function at the fixed point $\boldsymbol{\delta x} = 0$, giving for the covariance matrix $C_{\alpha\beta} = a_\alpha(0) a_\beta(0)$. As the outer product of a vector – with elements $a_\alpha(0)$ – with itself this has rank one and so is not invertible except in the case of a single observable, making the projection operator (2.2.4) ill-defined.

An orthogonal projection operator $\mathcal{Q} = 1 - \mathcal{P}$ is then defined. With the shorthand $a_\alpha(\boldsymbol{\delta x}, t) = a_\alpha(t)$, the projected equations are then written [27–29]

$$\frac{\partial}{\partial t} a_\alpha(t) = \sum_\beta a_\beta(t) \Omega_{\beta\alpha} + \int_0^t dt' \sum_\beta a_\beta(t') M_{\beta\alpha}(t - t') + r_\alpha(t) \tag{3.1.7}$$

The first term on the r.h.s. represents the rate matrix

$$\Omega_{\beta\alpha} = \sum_\gamma \left(\boldsymbol{C}^{-1}\right)_{\beta\gamma} (a_\gamma, \mathcal{L} a_\alpha) \tag{3.1.8}$$

and the second term is referred to as the memory function

$$M_{\beta\alpha}(\Delta t) = \sum_\gamma \left(\boldsymbol{C}^{-1}\right)_{\beta\gamma} \left(a_\gamma, \mathcal{L}\mathcal{Q} e^{\mathcal{Q}\mathcal{L}\mathcal{Q}\Delta t} \mathcal{Q}\mathcal{L} a_\alpha\right) \tag{3.1.9}$$

This represents the memory effects and determines how strongly the past values of observable a_β affect the present rate of change of a_α. Finally the third term is known as the random force

$$r_\alpha(t) = e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t} \mathcal{Q}\mathcal{L} a_\alpha. \tag{3.1.10}$$

Note that this notion of randomness does not imply that the random force resembles white noise as in e.g. Langevin equations. This is plausible given that it appears in the time evolution of the $a_\alpha(t)$, which are conditional averages over dynamical fluctuations.

In fact we will see later in (3.2.14) that the random force encodes primarily the initial conditions of the bulk variables.

The projected equations (3.1.7) as written are exact, and have several remarkable features. Firstly, they emphasize that memory terms must arise generically once we go from a description of the full system, in terms of $\boldsymbol{\delta x}$, to one in terms of a reduced number of observables. Secondly, they provide an "almost" closed set of equations for the chosen observables, with all non-autonomous effects collected in the random force term. Specifically, while the time evolution of each $a_\alpha(\boldsymbol{\delta x}, t)$ depends in principle on all details of the initial system state $\boldsymbol{\delta x}$, the projected equations (3.1.7) with the random force term omitted can be solved knowing only the initial values of the chosen observables, $a_\alpha(0)$.

To make use of the projected equations, we must be able to calculate the rate matrix and the memory functions, and say something about the statistics of the random force. Calculations of the rate matrix (3.1.8) and memory functions (3.1.9) are discussed in more detail in Section 3.2. Here we note only two useful identifies, which follow from the definitions of these quantities and that of the projection operator (2.2.4):

$$\begin{aligned} \sum_\beta a_\beta \Omega_{\beta\alpha} &= \mathcal{P}\mathcal{L}a_\alpha \\ \sum_\beta a_\beta M_{\beta\alpha}(\Delta t) &= \mathcal{P}\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}\mathcal{Q}\Delta t}\mathcal{Q}\mathcal{L}a_\alpha \end{aligned} \tag{3.1.11}$$

To find $\Omega_{\beta\alpha}$ and $M_{\beta\alpha}(\Delta t)$ we can then first evaluate the r.h.s. of these identities, and identify the coefficients of the different a_β.

As regards the statistics of the random force, there is a simple scenario where all correlation functions $\langle r_\gamma(t')r_\alpha(t)\rangle$ (for $t \geq t'$) can be deduced from the memory functions. This is the case where the operator $\mathcal{L}$ is self-adjoint with regards to the product (a, b), such that $(a, \mathcal{L}b) = (\mathcal{L}a, b)$ for any observables a and b. Using the fact that $\mathcal{Q}$ automatically has the same property one then finds [27, 28]

$$\begin{aligned} \sum_j C_{\gamma\beta}M_{\beta\alpha}(t+t') &= \left(\mathcal{Q}\mathcal{L}a_\gamma, e^{\mathcal{Q}\mathcal{L}\mathcal{Q}(t+t')}\mathcal{Q}\mathcal{L}a_\alpha\right) = \left(e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t'}\mathcal{Q}\mathcal{L}a_\gamma, e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t}\mathcal{Q}\mathcal{L}a_\alpha\right) \\ &= \langle r_\gamma(t')r_\alpha(t)\rangle \end{aligned} \tag{3.1.12}$$

Setting t or t' to zero shows that the memory function gives the correlations between the random force at the initial time and its value some time later. The self-adjointness of $\mathcal{L}$

required here normally holds in physical systems: these obey detailed balance, meaning that in the steady state there are no unbalanced probability fluxes. Protein interaction networks do not in general have this property[1], so that random force statistics have to be calculated separately. We therefore leave this matter as a point of investigation in Chapter 7, and note here only that the random force has the biological meaning of extrinsic noise acting on the subnetwork, arising from it being embedded in the bulk network.

What specific projected equations one obtains from the framework summarised above is of course largely dependent on the choice of observables a_α. This is discussed in more detail in Sec. 3.1.4. Here we just note that one useful convention is to employ observables with vanishing steady state average, $\langle a_\alpha \rangle \equiv (a_\alpha, 1) = 0$, which can always be achieved by subtracting any nonzero average from a_α. This convention has two benefits: first, it guarantees that the matrix $\boldsymbol{C}$ defined in (2.2.5) really is a correlation matrix for fluctuations around the steady state. Second, the projection operator then obeys $\mathcal{P}c = 0$, hence $\mathcal{Q}c = c$ and $\mathcal{Q}\mathcal{L}\mathcal{Q}c = \mathcal{Q}\mathcal{L}c = 0$. The operator $\mathcal{Q}\mathcal{L}\mathcal{Q}$ thus inherits from $\mathcal{L}$ the property that its application to any constant gives zero. As argued above for $\mathcal{L}$, this is equivalent to saying that the adjoint operator $(\mathcal{Q}\mathcal{L}\mathcal{Q})^{\mathrm{T}}$ conserves probability in the time evolution it generates. One can therefore think of the time-dependencies in the memory function and random force as resulting from a "projected evolution" of the system with this operator. In applications to physical systems, this is often used to argue that as a first approximation $\mathcal{Q}\mathcal{L}\mathcal{Q}$ can be replaced by $\mathcal{L}$ [30, 41], though this is not a path we follow here as we want to retain a quantitatively accurate projected description.

In order to evaluate the rate matrix (2.2.9) and memory functions (2.2.10) we have to calculate the various observable products (a, b) that occur, and from (2.2.6) these are defined in terms of the steady state distribution of $\boldsymbol{\delta x}$. In our case the latter is a vector of concentrations, shifted to zero mean. Our general strategy will be to consider suitably large reaction volumes so that the noise strength $\epsilon = 1/V$ is small. More specifically we require that for typical concentrations of any species, the absolute number of molecules

[1] Note also that even if detailed balance holds for a system described fully in terms of discrete numbers of molecules, it may be lost when going to our Kramers-Moyal expansion truncated at second order.

be large, say $Vy_i \gg 1$ for all i if we take the steady state concentrations as typical.[2] The steady state fluctuations $\boldsymbol{\delta x}$ will then be small, and we can find their distribution as the steady state to an approximate Fokker-Planck operator, obtained from $\mathcal{L}^{\mathrm{T}}$ by linearizing around $\boldsymbol{\delta x} = 0$. We emphasise that this simplification is used only for the steady state, and does not restrict the deviations from the steady state $\boldsymbol{\delta x}$ that can be considered in the projected equations, e.g. while the system evolves from some non-steady initial state.

In the linearised version of $\mathcal{L}$, the diffusion matrix $\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$ is evaluated at $\boldsymbol{\delta x} = 0$, i.e. at the steady state concentrations. The deterministic drift is linearised in $\boldsymbol{\delta x}$ so that it can be written in terms of a drift matrix $\boldsymbol{A}$ and a vector $\boldsymbol{\delta x}$ as $\boldsymbol{S}\boldsymbol{f} = \boldsymbol{A}\boldsymbol{\delta x}$. The steady state $P_{\mathrm{ss}}(\boldsymbol{\delta x})$ of such a Fokker-Planck operator is a Gaussian distribution for the $\boldsymbol{\delta x}$ with zero mean and a covariance matrix $\boldsymbol{\Sigma}$ that is a solution of the Lyapunov equation [44]

$$\boldsymbol{A}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \epsilon \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} = 0 \tag{3.1.13}$$

Once $\boldsymbol{\Sigma}$ is known, the inner products (2.2.6) in the projection can then be evaluated as Gaussian averages.

One proviso with this approach to finding $\boldsymbol{\Sigma}$, and hence $P_{\mathrm{ss}}(\boldsymbol{\delta x})$, is that the solution to the Lyapunov equation is not unique. This is because of the conservation laws: each fixed value of the conserved quantities leads to a different steady state distribution, and the generic solution for $\boldsymbol{\Sigma}$ represents a superposition of these distributions. In simple networks that we have analysed one particularly simple solution of this type is the one where each molecular species has independent Poisson fluctuations at steady state. This occurs for networks where the the conserved quantities are not fixed because diffusion overtakes the influence of fluctuations due to reactions [45]. For small ϵ the resulting product of Poisson distributions becomes a Gaussian with a diagonal covariance matrix $\boldsymbol{\Sigma}$. Because under Poisson statistics the variance of the number of molecules for each

[2] In the EGFR network discussed in Sec. 4, steady state concentrations range from 0.05 to 1000nMol [13]. If we estimate cells to have a diameter of 20μm and hence a volume of order $(20\mu\mathrm{m})^3$, this gives absolute steady state molecule numbers Vy_i in the range 240 to $4.8{\cdot}10^6$ and the criterion $Vy_i \gg 1$ is well satisfied. In separate large-scale studies in specific human cell lines [42, 43], protein abundances of up to $2{\cdot}10^7$ molecules per cell have been reported, with a median number across species of $1.8{\cdot}10^4$. The distribution of number of molecules is broad, but almost all (97.7%) species have more than 100 molecules per cell, so that a small noise approximation should again be justified.

species, Vx_i, equals its mean Vy_i, one has $\langle (V\delta x_i)^2 \rangle = Vy_i$ and hence $\Sigma_{ii} = \langle (\delta x_i)^2 \rangle = y_i/V = \epsilon y_i$. For brevity we will call such a covariance matrix "Poissonian".

The projection approach can be defined using different types of distribution; however we will see below that having a steady state distribution with Poissonian covariance matrix has a number of benefits. The main one is that the rate matrix terms in the projected equations will reproduce precisely those terms from the original evolution equations for the full network that describe reactions within the chosen subnetwork. The memory terms can then be interpreted directly as arising from the presence of the bulk. In view of this, we will use the Poissonian choice of covariance matrix throughout. This means that, depending on the network under study, our $P_{\rm ss}(\boldsymbol{\delta x})$ will only be an approximation of the true steady state distribution. However, this is not a serious obstacle: if one looks at the derivation [27] of the projected equations (3.1.7), one sees that in principle any distribution can be used to define the projection operator. (The exception is the detailed balance property discussed around (3.1.12), but we do not rely on this in our analysis.) The price we pay is that the random force is then "random", i.e. uncorrelated with the initial values of our chosen observables, under the Poisson distribution we have chosen, while under the true steady state distribution it will generally have nonzero correlations. This is a proviso that has to be born in mind, but it is easily outweighed by the fact that the projected equations will be simpler to interpret.

In summary, while the Poissonian covariance matrix assumption does represent a valid steady state in simple networks, more generally it should be viewed as an auxiliary construct that produces the simplest form of the projected equations for the subnetwork dynamics.

3.1.4 Choice of subnetwork observables

To calculate the projected equations we need to choose the set of observables $\{a_i(\boldsymbol{\delta x})\}$ that we will project on. We assume that a set of molecular species has been chosen as the subnetwork of interest, e.g. because of relevance to some biological function or experimental accessibility. As before we will call the rest of the species in the network the bulk. However, this division still leaves an element of choice as to which subnetwork observables to use in the projection. To illustrate the issues, we consider a small exam-

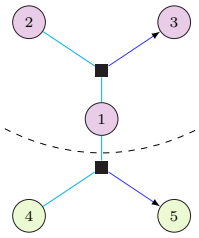


Figure 3.1: Sketch of a simple model protein interaction network. Protein 1 reacts with protein 2 to form complex 3, and in reverse 3 can dissociate into 1 and 2. There is an analogous reaction where 1 reacts with 4 to form 5, and the reverse dissociation. We choose the subnetwork to be species 1, 2 and 3 and the bulk to be 4 and 5, as indicated by the dashed line.

ple, represented graphically in Figure 3.1, with two complex formation and dissociation reactions as indicated below:

$$\begin{aligned} 1 + 2 &\underset{k^-_{3,12}}{\overset{k^+_{12,3}}{\rightleftharpoons}} 3 \\ 1 + 4 &\underset{k^-_{5,14}}{\overset{k^+_{14,5}}{\rightleftharpoons}} 5 \end{aligned} \tag{3.1.14}$$

The chemical Langevin equations for this network are

$$\begin{aligned} \frac{\partial}{\partial t}\delta x_1 &= k^-_{3,12}\delta x_3 - k^+_{12,3}\left(y_2\delta x_1 + y_1\delta x_2 + \delta x_1\delta x_2\right) \\ &\quad + k^-_{5,14}\delta x_5 - k^+_{14,5}\left(y_4\delta x_1 + y_1\delta x_4 + \delta x_1\delta x_4\right) + \eta_1 \\ \frac{\partial}{\partial t}\delta x_2 &= k^-_{3,12}\delta x_3 - k^+_{12,3}\left(y_2\delta x_1 + y_1\delta x_2 + \delta x_1\delta x_2\right) + \eta_2 \\ \frac{\partial}{\partial t}\delta x_3 &= k^+_{12,3}\left(y_2\delta x_1 + y_1\delta x_2 + \delta x_1\delta x_2\right) - k^-_{3,12}\delta x_3 + \eta_3 \end{aligned} \tag{3.1.15a}$$

$$\begin{aligned} \frac{\partial}{\partial t}\delta x_4 &= k^-_{5,14}\delta x_5 - k^+_{14,5}\left(y_4\delta x_1 + y_1\delta x_4 + \delta x_1\delta x_4\right) + \eta_4 \\ \frac{\partial}{\partial t}\delta x_5 &= k^+_{14,5}\left(y_4\delta x_1 + y_1\delta x_4 + \delta x_1\delta x_4\right) - k^-_{5,14}\delta x_5 + \eta_5 \end{aligned} \tag{3.1.15b}$$

Here the terms η_i are the contributions from the (intrinsic) noise. As in the general form of the adjoint Fokker-Planck operator (3.1.4), we have written concentration products $x_i x_j$ from the original mass action form (3.1.1) in terms of δx_i and δx_j and removed constant terms that cancel in steady state, giving $y_j\delta x_i + y_i\delta x_j + \delta x_i\delta x_j$.

We assume that the subnetwork of interest in this example consists of species 1, 2 and 3, and want to select observables for the projection method accordingly. The goal is to keep the number of variables small, for computational and conceptual expediency, while

retaining an explicit description of the subnetwork reactions (3.1.15a) in its original form.

As a first choice one could consider projecting onto only the *protein concentrations* in the subnetwork, $(\delta x_1, \delta x_2)$. Explicitly, this means we use only two observables, $a_1(\boldsymbol{\delta x}) = \delta x_1$ and $a_2(\boldsymbol{\delta x}) = \delta x_2$ where $\boldsymbol{\delta x} = (\delta x_1, \ldots, \delta x_5)^{\mathrm{T}}$. When we write down the projected equations (3.1.7), we should in principle write $a_1(t)$ and $a_2(t)$ and bear in mind that these are the conditional averages – over the stochastic noise from copy number fluctuations – of δx_1 and δx_2. However, as we are interested throughout in the *limit of small* ϵ, where the effect of averaging over the noise becomes negligible, we write directly δx_1 and δx_2.

Deferring for now a discussion of how rate matrix and memory functions are calculated in practice (see Secs. 3.2.1 and 3.2.2), we state directly the projected equation for δx_1 that results from the above choice of subnetwork observables:

$$\begin{aligned}\frac{\partial}{\partial t}\delta x_1 = &-k^+_{12,3}\left(y_2\delta x_1 + y_1\delta x_2\right) - k^+_{14,5}y_4\delta x_1 \\ &+ \int_0^t dt' \Big[k^+_{14,5}y_4(k^-_{5,14} + k^+_{14,5}y_1)e^{-(k^-_{5,14}+k^+_{14,5}y_1)(t-t')} \\ &\qquad +k^-_{3,12}k^+_{12,3}y_2 e^{-k^-_{3,12}(t-t')}\Big]\delta x_1(t') \\ &+ \int_0^t dt' \left[k^-_{3,12}k^+_{12,3}y_1 e^{-k^-_{3,12}(t-t')}\right]\delta x_2(t') + r_1(t)\end{aligned} \tag{3.1.16}$$

The terms from the rate matrix, which are the local-in-time contributions in the first line, are linear in δx_1 and δx_2 as expected from (3.1.7). They therefore do not capture all terms from the complex formation and dissociation reactions within the subnetwork, as written in the first line of (3.1.15a).

To include nonlinear terms, one could consider adding a third observable, $a_3 = \delta x_1\delta x_2 - \langle\delta x_1\delta x_2\rangle$, giving a projection onto *protein concentrations and products of protein concentrations*. We have written the subtraction of $\langle\delta x_1\delta x_2\rangle$ here for clarity to emphasize that also nonlinear observables must have zero mean, though for our chosen Poissonian steady state this average vanishes. The projected equation for δx_1 that results is similar to (3.1.16), but now explicitly includes the $\delta x_1\delta x_2$ term from the first line of (3.1.15a). However, the complex dissociation term $k^-_{3,12}\delta x_3$ still does not feature because the complex, species 3, remains "eliminated" from the subnetwork description. Its contribution is retained indirectly through the memory function, but not in a very transparent way.

The best option is therefore to project onto the *protein concentrations, products of pro-*

tein concentrations and complex concentrations from the subnetwork. With the vector of observables now $(a_1, \ldots, a_4) = (\delta x_1, \delta x_2, \delta x_3, \delta x_1 \delta x_2 - \langle \delta x_1 \delta x_2 \rangle)$, the projected equation for δx_1 becomes

$$\begin{aligned}\frac{\partial}{\partial t}\delta x_1 = {} & k^-_{3,12}\delta x_3 - k^+_{12,3}\left(y_2\delta x_1 + y_1\delta x_2 + \delta x_1\delta x_2\right) - k^+_{14,5}y_4\delta x_1 \\ & + \int_0^t dt' \left[M_{1,1}(t-t')\delta x_1(t') + M_{12,1}(t-t')\delta x_1(t')\delta x_2(t')\right] + r_1(t)\end{aligned} \tag{3.1.17}$$

All contributions relating to the subnetwork reaction $1 + 2 \rightleftharpoons 3$ now appear directly via the local-in-time rate matrix terms: compare the first line of (3.1.17) to (3.1.15a). The bulk, which here comprises just species 4 and 5, contributes an additional local-in-time term because of the reaction of 4 with 1. The other bulk effects are captured in the memory terms: as expected from our general discussion, feedback effects from the subnetwork into the bulk and back at a later time lead to the evolution of δx_1 being coupled linearly to its own history, via a "self memory" term; there is also a memory term that is nonlinear in concentration fluctuations. The linear self-memory function can be written explicitly as $M_{1,1}(t-t') = k^+_{14,5}y_4(k^-_{5,14} + k^+_{14,5}y_1)e^{-(k^-_{5,14}+k^+_{14,5}y_1)(t-t')}$; we omit the full expression for $M_{12,1}(t-t')$ for the sake of brevity. As expected, the reaction rates $k^+_{14,5}$ and $k^-_{5,14}$ relating to the bulk protein and complex that are being projected out from the description appear in the memory functions.

The upshot of the discussion so far is that we should project onto the concentrations of all molecular species from the subnetwork – both proteins and complexes – and all products of these concentrations. This gives projected equations where (a) all reactions taking place within the subnetwork are represented in their original form, as if the subnetwork was isolated, and (b) memory terms arise only from the presence of the bulk. One final choice left open here is which concentration products to include, only those occurring in the subnetwork reactions like $\delta x_1 \delta x_2$ above, or all possible concentration products (e.g. $\delta x_1 \delta x_2$, δx_1^2, δx_2^2, with nonzero averages subtracted as necessary). We will see below that the latter choice has advantages in a general treatment, in so far as it leads to smaller random force contributions.

3.1.5 Memory functions: initial orientation

We conclude this section by using the simple example above to provide some initial insights into the properties and intuitive meaning of memory functions.

We focus initially on the self-memory function $M_{1,1}(t-t')$. Figure 3.2 shows a sketch of this function, for a simple choice of reaction rates in appropriate dimensionless units. The self-memory function is positive, and decays exponentially with the time difference to the present. The sign implies that having a higher concentration of species 1 at some previous time t' ($\delta x_1(t') > 0$) will lead to more 1 being created at time t. To see why this is so, note that if more 1 is present at t', then more of species 5 will be created from the reaction with 4; this will then dissociate back into 1 at a later time, increasing the concentration of 1. This effect weakens as the concentration of 5 reverts to its steady state with time, explaining the decay of the memory function with the time difference $t-t'$.

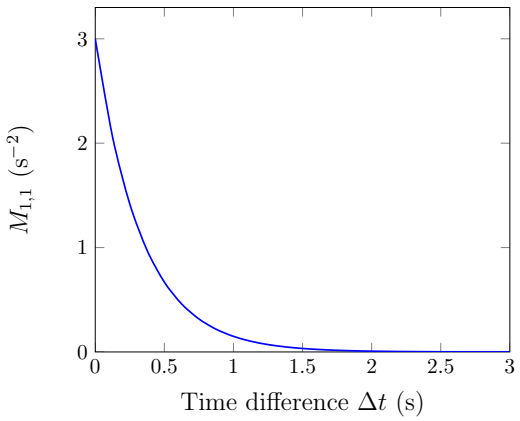


Figure 3.2: The coefficient of the self memory function for δx_1 with rate constants $k^+_{12,3} = k^+_{14,5} = 1$, $k^-_{3,12} = k^-_{5,14} = 2$, and steady state values $y_1 = y_2 = y_3 = 1$.

Looking at the self-memory function more quantitatively, one recognises that it reflects conservation laws of the proteins and complexes in the bulk, as it should. For the above example there is one such conservation law: the total concentration of species 4 and 5 is conserved, implying that deviations away from steady state are equal and opposite: $\delta x_4 = -\delta x_5$. Therefore the equation (3.1.15b) for the complex δx_5 can be rewritten as

$$\frac{\partial}{\partial t}\delta x_5 = k^+_{14,5}\left(y_4\delta x_1 - \delta x_1\delta x_5\right) - \left(k^+_{14,5}y_1 + k^-_{5,14}\right)\delta x_5 \tag{3.1.18}$$

If we now drop the $\delta x_1 \delta x_5$ term, which would contribute to the random force and to nonlinear memory terms and then integrate this equation we obtain

$$\delta x_5(t) = \int_0^t dt' k^+_{14,5} y_4 e^{-(k^-_{5,14}+k^+_{14,5}y_1)(t-t')} \delta x_1(t') \tag{3.1.19}$$

(up to an initial condition-dependent term which would give another contribution to the random force). Inserting into the equation for δx_1 in (3.1.15a) and using $\delta x_4 = -\delta x_5$ then gives the linear memory term in (3.1.17), showing that this accounts for the bulk conservation law as it should.

If we next look at the general structure of the memory terms in the projected equation for (3.1.17), we notice that in the linear memory terms only the history of δx_1 features, not that of δx_2. The same is true in the projected equation of motion for δx_2, which we have not given explicitly. As explained in more detail in Section 3.2.3, this is a general property of linear memory terms: the only variables that appear in these are the concentrations of "boundary species". Here a boundary species is one that has a direct reaction with a bulk species. In our example above, 1 is the only boundary species, while 2 and 3 are in the interior of the subnetwork. The intuitive reason why their histories do not appear in linear memory terms is that their effects on the bulk can only be "transmitted" indirectly via the time course of the concentration of species 1, rather than directly.

Finally we demonstrate the quantitative accuracy of the projected equations, i.e. (3.1.17) and the analogous equations for δx_2 and δx_3. We know that the equations are exact in the small noise limit $\epsilon \to 0$ that we have already taken, but the random force terms cannot be expressed in closed form, as discussed in more detail below. Our interest is therefore in assessing how accurate the projected subnetwork description is *when the random force terms are omitted but memory terms are retained.* Fig. 3.3 compares the solution of the resulting approximate projected equations with the solution of the full set of reaction equations (3.1.15). The two sets of time courses are visually indistinguishable, confirming that the projected subnetwork equations give a highly accurate description of the dynamics. The initial conditions were chosen so that concentrations of bulk species were at their steady state values. This is the regime where we expect the omitted random force terms to be smallest, as discussed in Sec. 4.4 below. We will also compare there to alternative reduced descriptions of subnetwork dynamics.

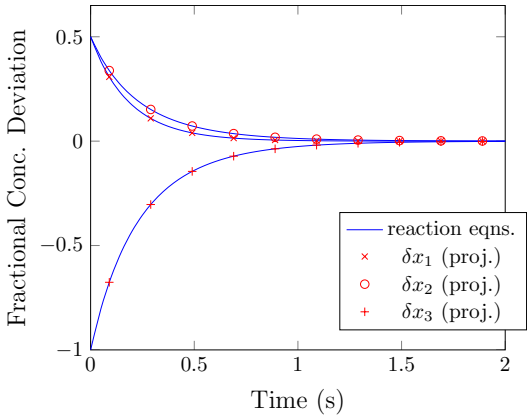


Figure 3.3: Concentration time courses for example (3.1.14), comparing the solution of the full reaction equations (3.1.15) (solid lines) and the solution of the subnetwork projected equations (3.1.17) (symbols). Note the excellent agreement even though random force terms were omitted from the projected equations. The y-axis shows fractional concentration deviations from the steady state, $\delta x_i/y_i$, so that a value of -1 corresponds to concentration $x_i = 0$. Rate constants $k^+_{12,3} = k^+_{14,5} = 1$ and $k^-_{3,12} = k^-_{5,14} = 2$; initial conditions $\delta x_1(0)/y_1 = \delta x_2(0)/y_2 = 1/2$, $\delta x_3(0)/y_3 = -1$, $\delta x_4(0) = \delta x_5(0) = 0$.

3.2 Memory functions: explicit expressions and general properties

In this section we give explicit expressions for memory functions describing the dynamics of protein interaction subnetworks. We study their general properties, in particular with a view to how they encode subnetwork-bulk interactions. In Sec. 3.2.1 we consider first a simplified scenario, where the dynamical equations of the original large network are linearised around the steady state. Applying the projection method to obtain a description of the subnetwork dynamics, the memory functions can be found explicitly; we validate the approach by comparing with the simpler approach of integrating out the bulk degrees of freedom directly. In Sec. 3.2.2 we then demonstrate that, more surprisingly, we can obtain the memory functions explicitly even for the full nonlinear dynamics. Throughout we focus on the small noise, large reaction volume limit $\epsilon \to 0$. Finally, in Sec. 3.2.3 we discuss some generic properties of memory functions.

3.2.1 Linearised dynamics

To get some insight into the general form of the projected equations we first consider a simplified problem, starting from a linearised description for the full network. The linearised reaction equations including copy number noise are

$$\partial_t \boldsymbol{\delta x} = \boldsymbol{A}\,\boldsymbol{\delta x} + \boldsymbol{\eta} \tag{3.2.1}$$

where $\boldsymbol{A}$ is as defined just before (3.1.13) and the covariance matrix $\epsilon \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$ of the noise η, which normally is $\boldsymbol{\delta x}$-dependent, is evaluated at steady state ($\boldsymbol{\delta x} = 0$). The corresponding adjoint Fokker-Planck operator is

$$\mathcal{L} = \sum_{ij} A_{ij}\delta x_j \frac{\partial}{\partial \delta x_i} + \frac{\epsilon}{2}\sum_{ij}(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}})_{ij}\frac{\partial^2}{\partial \delta x_i \partial \delta x_j} \tag{3.2.2}$$

In Section 3.1.4 we showed that in general, the most appropriate choice of subnetwork observables $\{a_i\}$ consists of the subnetwork concentrations and all their products. Now that we are considering linearised dynamics, we will only want to project onto the concentrations themselves, omitting the products. The linearised projected equations can then be written in the general form

$$\frac{\partial}{\partial t}\delta x_i(t) = \sum_{j=1}^{N^{\mathrm{s}}} \delta x_j(t)\Omega_{ji} + \int_0^t dt' \sum_{j=1}^{N^{\mathrm{s}}} \delta x_j(t') M_{ji}(t-t') + r_i(t) \tag{3.2.3}$$

and our aim will be to find explicit expressions for the rate matrix entries Ω_{ji} and the memory functions $M_{ji}(t-t')$. Note that, as it should be for a description of the subnetwork dynamics, the sums over j above run only over subnetwork concentrations. We assume here that these concentrations make up the first entries of the vector $\boldsymbol{\delta x}$, i.e. δx_j with $j = 1 \ldots N^{\mathrm{s}}$ where N^{s} is the number of subnetwork species. We will denote the subnetwork part of $\boldsymbol{\delta x}$ by $\boldsymbol{\delta x}^{\mathrm{s}}$, and the remaining bulk part by $\boldsymbol{\delta x}^{\mathrm{b}}$, so that $\boldsymbol{\delta x}^{\mathrm{T}} = (\boldsymbol{\delta x}^{\mathrm{sT}}, \boldsymbol{\delta x}^{\mathrm{bT}})$. Here T denotes the transpose of a column vector.

To find the rate matrix and memory functions from the general expressions (2.2.9) and (2.2.10), or equivalently (3.1.11), we need to be able to find the action of the operators $\mathcal{L}$, $\mathcal{P}$ and $\mathcal{Q}$ on the observables $a_i = \delta x_i$ ($i = 1, \ldots, N^{\mathrm{s}}$) and evaluate products of the form (a, b). Starting with the latter, we choose for the (approximate) steady-state distribution a Gaussian over $\boldsymbol{\delta x}$ with mean zero and Poissonian covariance matrix $\boldsymbol{\Sigma}$. The elements of this matrix then give the products $(\delta x_i, \delta x_j) = \Sigma_{ij}$. More specifically, if we partition

the covariance matrix depending on whether the relevant molecular species are in the subnetwork or bulk, as we did for the vector $\boldsymbol{\delta x}$, it can be written in the form

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^{\mathrm{s,s}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{\mathrm{b,b}} \end{pmatrix} \tag{3.2.4}$$

The Poissonian form for $\boldsymbol{\Sigma}$ forces zeros on the off-diagonal blocks as we have written. It also implies that $\boldsymbol{\Sigma}^{\mathrm{s,s}}$ and $\boldsymbol{\Sigma}^{\mathrm{b,b}}$ are diagonal, although we will not need this property in the following.

We can now write down the action of the projection operator (2.2.4) on a generic observable. For linear observables δx_i, which will be sufficient for our purposes, we obtain

$$\mathcal{P}\delta x_i = \sum_{j,k \leq N^{\mathrm{s}}} (\delta x_i, \delta x_j) \left(\boldsymbol{\Sigma}^{\mathrm{s,s}}\right)^{-1}_{jk} \delta x_k \tag{3.2.5}$$

Here we have used the fact that the observable correlation matrix $\boldsymbol{C}$, i.e. the covariance of the subnetwork concentrations, is just the top left block $\boldsymbol{\Sigma}^{\mathrm{s,s}}$ of $\boldsymbol{\Sigma}$. For $i = 1, \ldots, N^{\mathrm{s}}$ the first product is simply $\Sigma^{\mathrm{s,s}}_{ij}$ so that $\mathcal{P}\delta x_i = \delta x_i$. Conversely for $i = N^{\mathrm{s}} + 1, \ldots, N$ the product vanishes because of the block structure of $\boldsymbol{\Sigma}$, and $\mathcal{P}\delta x_i = 0$. If we collect these results, and the corresponding ones for the orthogonal projector $\mathcal{Q}$, in the form

$$\begin{aligned} \mathcal{P}\delta x_i &= \sum_{j=1}^{N} \delta x_j P_{ji} \\ \mathcal{Q}\delta x_i &= \sum_{j=1}^{N} \delta x_j Q_{ji} \end{aligned} \tag{3.2.6}$$

then the coefficients P_{ji} and Q_{ji} can be arranged into matrices with the simple block form

$$\boldsymbol{P} = \begin{pmatrix} \mathbb{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \boldsymbol{Q} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{1} \end{pmatrix} \tag{3.2.7}$$

These results are intuitively obvious: when we project onto the subnetwork, the only observables that survive are those from the subnetwork. Similarly when applying the orthogonal projection operator $\mathcal{Q}$, only bulk observables remain.

Finally we can also write the adjoint Fokker-Planck operator in a similar matrix form. Looking at (3.2.2), $\mathcal{L}\delta x_i = \sum_j A_{ij}\delta x_j$, so if we define

$$\mathcal{L}\delta x_i = \sum_{j=1}^{N} \delta x_j L_{ji} \tag{3.2.8}$$

then $\boldsymbol{L} = \boldsymbol{A}^{\mathrm{T}}$ is the transpose of the dynamical matrix. This makes sense as the equation of motion (2.2.3) for the conditionally averaged concentrations,

$$\frac{\partial}{\partial t}\delta x_i = \mathcal{L}\delta x_i = \sum_j \delta x_j L_{ji} \tag{3.2.9}$$

has to agree with the noise-averaged rate equation (3.2.1), $\partial_t \delta x_i = \sum_j A_{ij}\delta x_j = \sum_j \delta x_j (\boldsymbol{A}^{\mathrm{T}})_{ji}$. We can partition the matrix $\boldsymbol{L}$ representing the adjoint Fokker-Planck operator $\mathcal{L}$ into subnetwork and bulk contributions again, according to

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}^{\mathrm{s,s}} & \boldsymbol{L}^{\mathrm{s,b}} \\ \boldsymbol{L}^{\mathrm{b,s}} & \boldsymbol{L}^{\mathrm{b,b}} \end{pmatrix} \tag{3.2.10}$$

From the definition (3.2.8) one then sees that $\boldsymbol{L}^{\mathrm{b,s}}$, for example, contains the coefficients of bulk concentrations in the equations of motion for subnetwork concentrations.

Note that the matrix representations (3.2.6) and (3.2.8) have been defined so that the vector $\boldsymbol{\delta x}$ sits on the left, e.g. $\mathcal{P}\boldsymbol{\delta x}^{\mathrm{T}} = \boldsymbol{\delta x}^{\mathrm{T}}\boldsymbol{P}$. This has the advantage of maintaining the ordering of the matrices when operators are composed, for example $\mathcal{P}\mathcal{L}\delta x_i = \mathcal{P}\sum_j \delta x_j L_{ji} = \sum_{jk} \delta x_k P_{kj} L_{ji}$, or in vector form $\mathcal{P}\mathcal{L}\boldsymbol{\delta x}^{\mathrm{T}} = \boldsymbol{\delta x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{L}$.

This identity can now be employed directly to get the rate matrix terms in the projected equations (3.1.7). We use (3.1.11), i.e. $\sum_{j=1} \delta x_j \Omega_{ji} = \mathcal{P}\mathcal{L}\delta x_i$. This has to hold for all $i = 1, \ldots, N^{\mathrm{s}}$, so one reads off that $\boldsymbol{\Omega}$ is the top left block of $\boldsymbol{PL}$, which because of the simple form of (3.2.7) is simply the top left block of $\boldsymbol{L}$ in (3.2.10), i.e.

$$\boldsymbol{\Omega} = \boldsymbol{L}^{\mathrm{s,s}} \tag{3.2.11}$$

Similarly, the memory function obeys the identity (3.1.11):

$$\sum_{j=1}^{N^{\mathrm{s}}} \delta x_j M_{ji}(\Delta t) = \mathcal{P}\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}\mathcal{Q}\Delta t}\mathcal{Q}\mathcal{L}\delta x_i \tag{3.2.12}$$

Exploiting the correspondence between operators and matrices again, the r.h.s. is the i-th entry of the vector $\boldsymbol{\delta x}^{\mathrm{T}}\boldsymbol{PLQ}e^{\boldsymbol{QLQ}\Delta t}\boldsymbol{QL}$. Comparing with the l.h.s. shows that the matrix of memory functions $\boldsymbol{M}(\Delta t)$ is the top left block of $\boldsymbol{PLQ}e^{\boldsymbol{QLQ}\Delta t}\boldsymbol{QL}$, where $e^{\boldsymbol{QLQ}\Delta t}$ is a matrix exponential. Inserting the block decomposition (3.2.10) of $\boldsymbol{L}$ and exploiting again (3.2.7) shows that this can be written explicitly as

$$\boldsymbol{M}(\Delta t) = \boldsymbol{L}^{\mathrm{s,b}} e^{\boldsymbol{L}^{\mathrm{b,b}}\Delta t}\boldsymbol{L}^{\mathrm{b,s}} \tag{3.2.13}$$

With this and (3.2.11) we have obtained the desired explicit expressions for the rate and memory matrices of the projected equations (3.2.3). We note as an aside that, for the linearised scenario we are considering here, an expression for the random force can also be found. The definition (2.2.11) becomes $r_i(t) = e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t}\mathcal{Q}\mathcal{L}\delta x_i$, which is the i-th entry of the vector $\boldsymbol{\delta x}^{\mathrm{T}} e^{\boldsymbol{QLQ}t}\boldsymbol{QL}$. If we gather the required entries for $i = 1, \ldots, N^{\mathrm{s}}$ into a vector $\boldsymbol{r}(t)$, this can be simplified to

$$\boldsymbol{r}^{\mathrm{T}}(t) = \boldsymbol{\delta x}^{\mathrm{b\,T}} e^{\boldsymbol{L}^{\mathrm{b,b}}t}\boldsymbol{L}^{\mathrm{b,s}}. \tag{3.2.14}$$

We have gone through the application of the projection approach to the linearised dynamics to illustrate the steps involved in calculating the rate matrix and memory functions. For this relatively simple setup one can obtain the projected equations also more directly, by integrating out the bulk degrees of freedom. One starts from the equations of motion for the (conditionally averaged) concentrations, which read $\partial_t \boldsymbol{\delta x}^{\mathrm{T}} = \boldsymbol{\delta x}^{\mathrm{T}}\boldsymbol{L}$ or after decomposing into subnetwork and bulk terms

$$\partial_t \boldsymbol{\delta x}^{\mathrm{s\,T}} = \boldsymbol{\delta x}^{\mathrm{s\,T}}\boldsymbol{L}^{\mathrm{s,s}} + \boldsymbol{\delta x}^{\mathrm{b\,T}}\boldsymbol{L}^{\mathrm{b,s}} \tag{3.2.15a}$$

$$\partial_t \boldsymbol{\delta x}^{\mathrm{b\,T}} = \boldsymbol{\delta x}^{\mathrm{s\,T}}\boldsymbol{L}^{\mathrm{s,b}} + \boldsymbol{\delta x}^{\mathrm{b\,T}}\boldsymbol{L}^{\mathrm{b,b}} \tag{3.2.15b}$$

The solution for the bulk concentrations reads

$$\boldsymbol{\delta x}^{\mathrm{b\,T}}(t) = \boldsymbol{\delta x}^{\mathrm{b\,T}}(0)e^{\boldsymbol{L}^{\mathrm{b,b}}t} + \int_0^t dt'\, \boldsymbol{\delta x}^{\mathrm{s\,T}}(t')\boldsymbol{L}^{\mathrm{s,b}}e^{\boldsymbol{L}^{\mathrm{b,b}}(t-t')} \tag{3.2.16}$$

and substituting into the first line of (3.2.15a) gives for the subnetwork concentrations

$$\partial_t \boldsymbol{\delta x}^{\mathrm{s\,T}}(t) = \boldsymbol{\delta x}^{\mathrm{s\,T}}\boldsymbol{L}^{\mathrm{s,s}} + \int_0^t dt'\, \boldsymbol{\delta x}^{\mathrm{s\,T}}(t')\boldsymbol{L}^{\mathrm{s,b}}e^{\boldsymbol{L}^{\mathrm{b,b}}(t-t')}\boldsymbol{L}^{\mathrm{b,s}} + \boldsymbol{\delta x}^{\mathrm{b\,T}}(0)e^{\boldsymbol{L}^{\mathrm{b,b}}t}\boldsymbol{L}^{\mathrm{b,s}} \tag{3.2.17}$$

which is exactly (3.2.3) with the rate matrix (3.2.11), memory matrix (3.2.13) and random force (3.2.14). This derivation shows explicitly how memory arises from bulk degrees of freedom being influenced by past behaviour of the subnetwork, and then feeding this influence back to the subnetwork at a later time. One also sees either from this or from (3.2.14) that the random force terms account for the effects of potentially unknown bulk initial conditions $\boldsymbol{\delta x}^{\mathrm{b}}(0)$. When $\boldsymbol{\delta x}^{\mathrm{b}}(0) = 0$, i.e. when the bulk is initially in steady state, then the random force vanishes. The solution of the projected equations for the subnetwork with the random force omitted will then match exactly the solution of the original linearised reaction equations (3.2.9). This motivates the good agreement

we observed between the two sets of solution in the simple example of Sec. 3.1.4, cf. Fig. 3.1, although there we were dealing with the full nonlinear reaction equations. This is the topic we consider next.

3.2.2 Nonlinear dynamics

The projection approach as examplified for linearised dynamics in the previous section may seem formal and somewhat indirect, given that bulk degrees of freedom can be eliminated directly. The method comes into its own, however, when we consider the full nonlinear reaction equations (3.1.1), where a direct elimination approach is not feasible. We show in this section that, non-trivially for a nonlinear case, explicit expressions for the rate matrix and memory functions in the projected equations can be found. We will appeal to the small noise limit $\epsilon \to 0$ as before, and will need to examine carefully what terms survive in this limit. Note that this was not necessary for the linearised dynamics, where the noise drops out from the equations for the conditionally averaged concentrations, whatever the value of ϵ. Guided by the discussion of the linear case, we will again aim to find a suitable matrix representation for the operators involved.

Regarding the choice of observables $\{a_\alpha\}$ for the projection we follow the discussion in Section 3.1.4 and include the concentrations of the subnetwork species and all products of these concentrations, shifted to zero mean as necessary. The list of observables is then $\boldsymbol{a}^{\mathrm{T}} = (\delta x_1, \ldots, \delta x_{N^{\mathrm{s}}}, \delta x_1^2 - \langle \delta x_1^2 \rangle, \delta x_1 \delta x_2 - \langle \delta x_1 \delta x_2 \rangle, \ldots, \delta x_{N^{\mathrm{s}}}^2 - \langle \delta x_{N^{\mathrm{s}}}^2 \rangle)$, containing in total $N^{\mathrm{s}} + N^{\mathrm{s}}(N^{\mathrm{s}}+1)/2 = N^{\mathrm{s}}(N^{\mathrm{s}}+3)/2$ distinct quantities. For the steady state distribution we take again a zero mean Gaussian for $\boldsymbol{\delta x}$, with a Poissonian covariance matrix. The steady state averages $\langle \delta x_i \delta x_j \rangle$ are then $\mathcal{O}(\epsilon)$ and can be neglected against terms of order unity. Applying this simplification, the nonlinear projected equations for the subnetwork concentrations then follow from the general result (3.1.7) as

$$\begin{aligned}\frac{\partial}{\partial t}\delta x_i = &\sum_{j=1}^{N^{\mathrm{s}}} \delta x_j \Omega_{ji}^{\mathrm{s,s}} + \sum_{1 \le j \le k \le N^{\mathrm{s}}} \delta x_j \delta x_k \Omega_{(jk)i}^{\mathrm{ss,s}} \\ &+ \int_0^t dt' \left(\sum_{j=1}^{N^{\mathrm{s}}} \delta x_j(t') M_{ji}^{\mathrm{s,s}}(t-t') + \sum_{1 \le j \le k \le N^{\mathrm{s}}} \delta x_j(t') \delta x_k(t') M_{(jk)i}^{\mathrm{ss,s}}(t-t') \right) \\ &+ r_i(t)\end{aligned} \tag{3.2.18}$$

Here we have split sums over observables into linear and nonlinear observables to show the structure more clearly. Accordingly there are now linear and nonlinear rate matrix and memory contributions. The bracket on the subscript in the nonlinear terms $\Omega^{\mathrm{ss,s}}_{(jk)i}$ and $M^{\mathrm{ss,s}}_{(jk)i}$ indicates the effect of a concentration product $\delta x_j \delta x_k$ on the time evolution of δx_i. As before we have not distinguished in our notation between the original concentration variables δx_i or $\delta x_i \delta x_j$ and their averages conditional on a given initial set of concentrations across the network, because the two become identical for $\epsilon \to 0$. Finally, all indices relate only to subnetwork variables and so lie in the range $1, \ldots, N^{\mathrm{s}}$.

Our goal in this section is to find explicit expressions for the linear and nonlinear rate matrix and memory functions in (3.2.18). To establish whether we can achieve this using matrix representations of the relevant operators, we first look at the terms we obtain by applying the operators $\mathcal{L}, \mathcal{P}$ and $\mathcal{Q}$ to concentrations and products of concentrations. The adjoint Fokker-Planck operator $\mathcal{L}$ from (3.1.4) contains single derivatives for the deterministic drift terms, multiplied by terms of order δx and δx^2, and second derivatives for the diffusion terms from copy number noise. The latter come with a factor ϵ and are multiplied by elements of the matrix $\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$. From (2.1.7) these get their δx-dependence from the reaction fluxes $\boldsymbol{f}$ and thus contain terms of up to quadratic order in δx. Applying then $\mathcal{L}$ to any linear concentration observable gives

$$\mathcal{L}\delta x_i = \delta x + \delta x^2 \tag{3.2.19}$$

because the diffusion piece does not contribute. The symbolic shorthand on the r.h.s. indicates a linear combination of terms of the form δx_i and $\delta x_j \delta x_k$. The analogous statement for a product of concentrations is

$$\mathcal{L}\delta x_i \delta x_j = \delta x^2 + \delta x^3 + \mathcal{O}(\epsilon) \tag{3.2.20}$$

because in the deterministic piece of $\mathcal{L}$ the first derivative now leaves one power of δx. The terms generated by the diffusion part are of order ϵ, $\epsilon\,\delta x$ and $\epsilon\,\delta x^2$, and we denote such terms summarily by $\mathcal{O}(\epsilon)$. To summarize the last two relations, define $\boldsymbol{z}$ as a vector containing all concentrations δx_i from the entire network as well as all concentration products $\delta x_j \delta x_k$: $\boldsymbol{z}^{\mathrm{T}} = (\delta x_1, \ldots, \delta x_N, \delta x_1^2 - \langle \delta x_1^2 \rangle, \delta x_1 \delta x_2 - \langle \delta x_1 \delta x_2 \rangle, \ldots, \delta x_N^2 - \langle \delta x_N^2 \rangle)$. Note that this is different from the vector $\boldsymbol{a}$, which only contains the elements of $\boldsymbol{z}$ that relate exclusively to the subnetwork. We can now write (3.2.19) and (3.2.20) together

in the form

$$\mathcal{L}z_\alpha = \sum_\beta z_\beta L_{\beta\alpha} + \delta x^3 + \mathcal{O}(\epsilon) \tag{3.2.21}$$

where α and β lie in the range $1, \ldots, N(N+3)/2$ and $L_{\beta\alpha}$ is a suitably defined matrix. Finally we have for the action of $\mathcal{L}$ on an n-th order product of concentrations

$$\mathcal{L}\delta x^n = \delta x^n + \delta x^{n+1} + \mathcal{O}(\epsilon), \quad n \geq 3 \tag{3.2.22}$$

where the first two terms on the r.h.s. again come from the deterministic drift.

The projection operators $\mathcal{P}$ and $\mathcal{Q}$ have similar properties as we show next. From the definition (2.2.4), we need the correlations (a_α, a_β) to get the projector. Fortunately, these are diagonal for our choice of a Poissonian covariance matrix $\mathbf{\Sigma}$. Firstly, because odd moments of a zero mean Gaussian vanish, there are no correlations between linear and quadratic observables. Correlations among linear observables are diagonal as before, $(\delta x_i, \delta x_j) = \Sigma_{ij} = \epsilon y_i \delta_{ij}$. The correlation among quadratic variables can be worked out using Wick's theorem [46]

$$\begin{aligned}(\delta x_i \delta x_j - \langle \delta x_i \delta x_j \rangle, \delta x_k \delta x_l - \langle \delta x_k \delta x_l \rangle) &= \langle \delta x_i \delta x_j \delta x_k \delta x_l \rangle - \langle \delta x_i \delta x_j \rangle \langle \delta x_k \delta x_l \rangle \\ &= \langle \delta x_i \delta x_k \rangle \langle \delta x_j \delta x_l \rangle + \langle \delta x_i \delta x_l \rangle \langle \delta x_j \delta x_k \rangle\end{aligned} \tag{3.2.23}$$

The surviving first term is nonzero only if $i = k$ and $j = l$, and similarly for the second one. Taking the indices as ordered ($i \leq j$ and $k \leq l$) then shows that the only nonzero correlations are the diagonal ones:

$$(\delta x_i \delta x_j - \langle \delta x_i \delta x_j \rangle, \delta x_i \delta x_j - \langle \delta x_i \delta x_j \rangle) = (1 + \delta_{ij}) \langle \delta x_i^2 \rangle \langle \delta x_j^2 \rangle = (1 + \delta_{ij}) \epsilon^2 y_i y_j \tag{3.2.24}$$

where the δ_{ij} accounts for the fact that the last term in (3.2.23) contributes only when $i = j$.

The projection operator now becomes, if we collect the above results for the observable correlation matrix $C_{\alpha\beta} = (a_\alpha, a_\beta)$ and split the sum over observables in the general definition (2.2.4) into linear and nonlinear terms,

$$\mathcal{P}b = \sum_{i=1}^{N^s} \delta x_i \frac{\langle \delta x_i b \rangle}{\epsilon y_i} + \sum_{i=1}^{N^s} (\delta x_i^2 - \epsilon y_i) \frac{\langle (\delta x_i^2 - \epsilon y_i) b \rangle}{2\epsilon^2 y_i^2} + \sum_{1 \leq i < j \leq N^s} \delta x_i \delta x_j \frac{\langle \delta x_i \delta x_j b \rangle}{\epsilon^2 y_i y_j} \tag{3.2.25}$$

For linear observables it follows that $\mathcal{P}\delta x_i = \delta x_i$ for subnetwork concentrations ($i = 1, \ldots, N^s$) and $= 0$ for bulk concentrations ($i > N^s$). Similarly, $\mathcal{P}\delta x_i \delta x_j = \delta x_i \delta x_j$ when

both indices are in the subnetwork range, and $= 0$ otherwise. The only exception is the case of two equal indices ($i = j$) in the subnetwork, where $\mathcal{P}\delta x_i^2 = \delta x_i^2 - \epsilon y_i$. Using again the vector $\boldsymbol{z}$ collecting all linear and quadratic observables from the network this means there is a matrix $\boldsymbol{P}$, given explicitly in (3.2.33) below, such that

$$\mathcal{P}z_\alpha = \sum_\beta z_\beta P_{\beta\alpha} + \mathcal{O}(\epsilon) \tag{3.2.26}$$

We also need to know the action of $\mathcal{P}$ on higher order observables $b = \delta x^n$ with $n \geq 3$. If n is odd, then only the linear terms in (3.2.25) contribute, with $\langle \delta x_i b \rangle$ proportional to $\epsilon^{(n+1)/2}$ from the scaling of the covariances. Thus $\mathcal{P}\delta x^n$ is of order $\epsilon^{(n-1)/2}\delta x$, which is $\mathcal{O}(\epsilon)$ as $n \geq 3$. For even n, we get only the quadratic terms from (3.2.25); the products with b are proportional to $\epsilon^{n/2+1}$ and so $\mathcal{P}\delta x^n$ is order $\epsilon^{n/2-1}[\delta x^2 + \mathcal{O}(\epsilon)]$, hence again $\mathcal{O}(\epsilon)$ as the smallest even value of n is 4. Thus

$$\mathcal{P}\delta x^n = \mathcal{O}(\epsilon), \quad n \geq 3 \tag{3.2.27}$$

The analogous properties of the orthogonal projector $\mathcal{Q}$ follow directly from the definition $\mathcal{Q}b = b - \mathcal{P}b$: its action on linear or quadratic observables, can be represented by a matrix,

$$\mathcal{Q}z_\alpha = \sum_\beta z_\beta Q_{\beta\alpha} \tag{3.2.28}$$

while higher order terms remain of higher order:

$$\mathcal{Q}\delta x^n = \delta x^n + \mathcal{O}(\epsilon), \quad n \geq 3 \tag{3.2.29}$$

We can now summarise the matrix representations for the nonlinear case. These matrices are defined by the action of the operators on linear or quadratic observables, up to terms that are at least cubic in concentration or proportional to ϵ:

$$\begin{aligned} \mathcal{L}z_\alpha &= \sum_\beta z_\beta L_{\beta\alpha} + \delta x^3 + \mathcal{O}(\epsilon) \\ \mathcal{P}z_\alpha &= \sum_\beta z_\beta P_{\beta\alpha} + \delta x^3 + \mathcal{O}(\epsilon) \\ \mathcal{Q}z_\alpha &= \sum_\beta z_\beta Q_{\beta\alpha} + \delta x^3 + \mathcal{O}(\epsilon) \end{aligned} \tag{3.2.30}$$

On the other hand for higher order observables, only terms of the same or higher order are created, or ones proportional to ϵ:

$$\mathcal{L}\delta x^n = \mathcal{O}(\delta x^n) + \mathcal{O}(\epsilon), \quad n \geq 3 \tag{3.2.31}$$

and similarly for $\mathcal{P}$ and $\mathcal{Q}$. Terms of order ϵ also remain of order ϵ or higher when one of the three operators is applied. It then follows that, as in the linear case, the product (composition) of any two operators has the same properties, and its matrix representation is the product of the matrices for the two operators. To see this consider e.g.

$$\mathcal{L}\mathcal{Q}z_\alpha = \mathcal{L}\Big(\sum_\beta z_\beta Q_{\beta\alpha} + \mathcal{O}(\delta x^3) + \mathcal{O}(\epsilon)\Big) = \sum_{\beta,\gamma} z_\gamma L_{\gamma\beta} Q_{\beta\alpha} + \mathcal{O}(\delta x^3) + \mathcal{O}(\epsilon) \quad (3.2.32)$$

This is the key result, which extends by induction to products over any number of operators.

It will be useful later to have the explicit forms of the nonlinear matrix representations:

$$\boldsymbol{P} = \begin{pmatrix} \mathbb{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}^{\mathrm{s,s}} & \boldsymbol{L}^{\mathrm{s,b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{L}^{\mathrm{b,s}} & \boldsymbol{L}^{\mathrm{b,b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{L}^{\mathrm{ss,s}} & \boldsymbol{L}^{\mathrm{ss,b}} & \boldsymbol{L}^{\mathrm{ss,ss}} & \boldsymbol{L}^{\mathrm{ss,sb}} & \boldsymbol{L}^{\mathrm{ss,bb}} \\ \boldsymbol{L}^{\mathrm{sb,s}} & \boldsymbol{L}^{\mathrm{sb,b}} & \boldsymbol{L}^{\mathrm{sb,ss}} & \boldsymbol{L}^{\mathrm{sb,sb}} & \boldsymbol{L}^{\mathrm{sb,bb}} \\ \boldsymbol{L}^{\mathrm{bb,s}} & \boldsymbol{L}^{\mathrm{bb,b}} & \boldsymbol{L}^{\mathrm{bb,ss}} & \boldsymbol{L}^{\mathrm{bb,sb}} & \boldsymbol{L}^{\mathrm{bb,bb}} \end{pmatrix} \quad (3.2.33)$$

Here the five rows and columns of the block structure relate to: linear subnetwork observables (s), linear bulk observables (b), product of subnetwork concentrations (ss), mixed subnetwork-bulk products of concentrations (sb) and products of bulk concentrations (bb). The fact that the top right blocks of $\boldsymbol{L}$ vanish comes from the statement (3.2.20): application of $\mathcal{L}$ to quadratic observables does not give linear terms. The matrix representation $\boldsymbol{Q}$ of $\mathcal{Q}$ is analogous to that of $\boldsymbol{P}$, with the roles of zero and identity matrices in the diagonal blocks interchanged.

We can proceed at this point to find the rate matrix for the nonlinear projected equations. Using (3.1.11), we need to apply first the adjoint Fokker-Planck operator $\mathcal{L}$ to an observable, and then the projector $\mathcal{P}$. The matrix representation of this product of operators is $\boldsymbol{PL}$, thus

$$\mathcal{P}\mathcal{L}z_\alpha = (\boldsymbol{z}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{L})_\alpha + \mathcal{O}(\epsilon) \quad (3.2.34)$$

where there are no $\mathcal{O}(\delta x^3)$ terms because the projector satisfies not just (3.2.30) but in fact the stronger (3.2.27). The $\mathcal{O}(\epsilon)$ term can furthermore be dropped when $\epsilon \to 0$. We now only need to choose for α the indices that give us the subnetwork entries from $\boldsymbol{z}$, and can then read from (3.1.11) the rate matrix entries. The relevant indices are

those in the first and third block columns of the matrices in (3.2.33). Writing out those columns of $\boldsymbol{PL}$ shows that the linear rate matrix, whose elements are written $\Omega_{ji}^{\mathrm{s,s}}$ in the projected equations (3.2.18), is simply the block $\boldsymbol{L}^{\mathrm{s,s}}$ of $\boldsymbol{L}$:

$$\boldsymbol{\Omega}^{\mathrm{s,s}} = \boldsymbol{L}^{\mathrm{s,s}} \tag{3.2.35}$$

As one might have expected, this is the same result as for the linearised dynamics discussed in Sec. 3.2.1. Similarly the nonlinear rate matrix is the block

$$\boldsymbol{\Omega}^{\mathrm{ss,s}} = \boldsymbol{L}^{\mathrm{ss,s}} \tag{3.2.36}$$

The same logic can now be applied to the calculation of the linear and nonlinear memory functions, starting from (3.1.11). The required operator involves an operator exponential, which can be expressed as a series:

$$\mathcal{PLQ}e^{\mathcal{QLQ}t}\mathcal{QL} = \sum_{n=0}^{\infty} \frac{t^n}{n!} \mathcal{PLQ}(\mathcal{QLQ})^n \mathcal{QL} \tag{3.2.37}$$

Every term in the sum is a product of operators, whose matrix representation will be the product of the matrices for the individual operators. Adding the series back up gives a matrix exponential, so that

$$\mathcal{PLQ}e^{\mathcal{QLQ}t}\mathcal{QL}z_\alpha = (\boldsymbol{z}^{\mathrm{T}}\boldsymbol{PLQ}e^{\boldsymbol{QLQ}t}\boldsymbol{QL})_\alpha + \mathcal{O}(\epsilon) \tag{3.2.38}$$

As before $\mathcal{O}(\delta x^3)$ terms are absent because the leftmost operator is the projector $\mathcal{P}$, and we can drop the $\mathcal{O}(\epsilon)$ terms when $\epsilon \to 0$. The remainder of the reasoning is as for the rate matrix: the linear memory functions $M_{ji}^{\mathrm{s,s}}(t)$ are the elements of the top left (s,s) block of $\boldsymbol{PLQ}e^{\boldsymbol{QLQ}t}\boldsymbol{QL}$, while the nonlinear memory functions $M_{(jk)i}^{\mathrm{ss,s}}(t)$ are those of the (ss,s) block.

Also for the memory functions one can show that the linear contributions are the same as for the linearised dynamics. To see this, one can write the building blocks

of $\boldsymbol{PLQ}e^{\boldsymbol{QLQ}t}\boldsymbol{QL}$ in block form:

$$\boldsymbol{PL} = \begin{pmatrix} \boldsymbol{L}^{\mathrm{s,s}} & \boldsymbol{L}^{\mathrm{s,b}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}^{\mathrm{ss,s}} & \boldsymbol{L}^{\mathrm{ss,b}} & \boldsymbol{L}^{\mathrm{ss,ss}} & \boldsymbol{L}^{\mathrm{ss,sb}} & \boldsymbol{L}^{\mathrm{ss,bb}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{QLQ} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}^{\mathrm{b,b}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}^{\mathrm{sb,b}} & \boldsymbol{0} & \boldsymbol{L}^{\mathrm{sb,sb}} & \boldsymbol{L}^{\mathrm{sb,bb}} \\ \boldsymbol{0} & \boldsymbol{L}^{\mathrm{bb,b}} & \boldsymbol{0} & \boldsymbol{L}^{\mathrm{bb,sb}} & \boldsymbol{L}^{\mathrm{bb,bb}} \end{pmatrix} \tag{3.2.39}$$

$$\boldsymbol{QL} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}^{\mathrm{b,s}} & \boldsymbol{L}^{\mathrm{b,b}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}^{\mathrm{sb,s}} & \boldsymbol{L}^{\mathrm{sb,b}} & \boldsymbol{L}^{\mathrm{sb,ss}} & \boldsymbol{L}^{\mathrm{sb,sb}} & \boldsymbol{L}^{\mathrm{sb,bb}} \\ \boldsymbol{L}^{\mathrm{bb,s}} & \boldsymbol{L}^{\mathrm{bb,b}} & \boldsymbol{L}^{\mathrm{bb,ss}} & \boldsymbol{L}^{\mathrm{bb,sb}} & \boldsymbol{L}^{\mathrm{bb,bb}} \end{pmatrix}$$

If we momentarily think of the sb and bb columns and rows as one, denoted "∗b"below, then $\boldsymbol{QLQ}$ has a lower triangular block structure. It follows that $e^{\boldsymbol{QLQ}t}$ has the same structure, with the diagonal blocks the exponentials of those of $\boldsymbol{QLQ}$. In particular the (b,b) block of $e^{\boldsymbol{QLQ}t}$ is $e^{\boldsymbol{L}^{\mathrm{b,b}}t}$. Multiplying by $\boldsymbol{PL}$ and $\boldsymbol{QL}$ from the left and right respectively, one then finds a matrix with (s, s) block equal to

$$\boldsymbol{M}^{\mathrm{s,s}}(\Delta t) = \boldsymbol{L}^{\mathrm{s,b}} e^{\boldsymbol{L}^{\mathrm{b,b}}\Delta t}\boldsymbol{L}^{\mathrm{b,s}} \tag{3.2.40}$$

in agreement with the result from the linearised dynamics.

The nonlinear memory matrix can be obtained similarly with a bit of algebra. Writing $\boldsymbol{E}(\Delta t) = e^{\boldsymbol{QLQ}\Delta t}$, the result has the form

$$\boldsymbol{M}^{\mathrm{ss,s}}(\Delta t) = \boldsymbol{L}^{\mathrm{ss,b}}\boldsymbol{E}^{\mathrm{b,b}}(\Delta t)\boldsymbol{L}^{\mathrm{b,s}} + \boldsymbol{L}^{\mathrm{ss,*b}}\boldsymbol{E}^{\mathrm{*b,b}}(\Delta t)\boldsymbol{L}^{\mathrm{b,s}} + \boldsymbol{L}^{\mathrm{ss,*b}}\boldsymbol{E}^{\mathrm{*b,*b}}(\Delta t)\boldsymbol{L}^{\mathrm{*b,s}} \tag{3.2.41}$$

We comment finally on the random force terms $r_i(t)$ in the nonlinear projected equations (3.2.18) for the subnetwork dynamics. From (2.2.11) we have $r_i(t) = e^{\mathcal{QLQ}t}\mathcal{QL}\delta x_i$. Given that the δx_i make up the first N^{s} components of the vector $\boldsymbol{z}$, we apply the same

argument as for the memory function:

$$e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t}\mathcal{Q}\mathcal{L}z_i = (\boldsymbol{z}^{\mathrm{T}}e^{\boldsymbol{QLQ}t}\boldsymbol{QL})_i + \mathcal{O}(\delta x^3) + \mathcal{O}(\epsilon) \tag{3.2.42}$$

For $\epsilon \to 0$ the last term can again the dropped, but the $\mathcal{O}(\delta x^3)$ terms remain as we do not have a projection operator $\mathcal{P}$ applied last that would remove them. The random force can therefore not be calculated in closed form from the matrix representations we have introduced.

The linear and quadratic contributions to the random force $r_i(t)$ are known explicitly, and given by the i-th entry of $\boldsymbol{z}^{\mathrm{T}}e^{\boldsymbol{QLQ}t}\boldsymbol{QL}$. We look briefly at which concentrations δx_i enter here. Expanding out the matrix exponential, one sees that all terms in $e^{\boldsymbol{QLQ}t}\boldsymbol{QL}$ contain $\boldsymbol{Q}$ as the leftmost factor. From (3.2.33) together with $\boldsymbol{Q} = \mathbb{1} - \boldsymbol{P}$, only the block rows labelled b, sb and bb of this matrix are nonzero. Thus $\boldsymbol{z}^{\mathrm{T}}e^{\boldsymbol{QLQ}t}\boldsymbol{QL}$ involves only linear terms δx_i from the bulk, and quadratic terms $\delta x_j \delta x_k$ with one or both factors in the bulk. All linear and quadratic terms in the random force therefore vanish when the bulk initial conditions are at steady state, $\delta x_i = 0$ for $i > N^{\mathrm{s}}$. Only third order and higher terms remain, so we expect the random force to be small or negligible in this case, consistent with the results of Fig. 3.3 above.

We can now also see why it is useful to include *all* products of subnetwork concentrations among our set of observables for the projection. These products are then removed from the random force by the orthogonal projector $\boldsymbol{Q}$, ensuring it vanishes to linear and quadratic order for a bulk initially in steady state. If we choose only to include *some* subnetwork concentration products, e.g. those that appear in the original reaction equations (3.1.1), then the remaining ones can and generically will appear in the random force, giving non-vanishing quadratic random force terms even for an initial bulk steady state. As we normally want to use the projected equations in the approximated form where random force terms are omitted, including all subnetwork products in the projection is preferable as it will lead to smaller random force contributions.

3.2.3 Properties of memory functions

We next discuss some generic properties of memory functions, based on the explicit expressions for linear and nonlinear memory functions obtained in the previous section.

Considering which memory functions are nonzero shows that the nonzero memory functions appear in the equations for molecular species on the *boundary* of the subnetwork to the bulk (Sec. 3.2.3.1). For the nonzero memory functions we then discuss what sets their amplitude, i.e. the value at zero time difference (Sec. 3.2.3.2) and the timescale of their decay as this time difference increases (Sec. 3.2.3.3).

3.2.3.1 Boundary structure

Before discussing which memory functions can be nonzero, we need to agree some conventions for how molecular species can be divided between a subnetwork and bulk. We will assume that a subnetwork complex can only be created by two subnetwork proteins, whereas a bulk complex can be created by either two bulk proteins or a bulk and a subnetwork protein. This is a reasonable biological assumption: we are generally interested in subnetworks that are small, e.g. to aid interpretability of the dynamics, and contain molecular species that are well understood in the sense that they do not form "unknown" complexes that we would assign to the bulk. Similarly for complexes retained in the subnetwork description we can expect that it is known how they are formed, and that the constituent proteins are included in the subnetwork.

To see the consequences for the (nonlinear) matrix representation of the adjoint Fokker-Planck operator in (3.2.33), recall that the equation of motion for linear and quadratic observables is, from (2.2.3) and (3.2.21), $\partial_t z_\alpha = \sum_\beta z_\beta L_{\beta\alpha} + \delta x^3 + \mathcal{O}(\epsilon)$. Thus the second index in $L_{\beta\alpha}$ determines which equation of motion we are considering, while the first index labels the variables featuring in this equation. Our assumptions above then mean that some of the blocks of $\boldsymbol{L}$ are zero. This applies to $\boldsymbol{L}^{\mathrm{ss,b}}$, which encodes contributions quadratic in subnetwork concentrations to the equation of motion for a bulk concentration. Looking back at the equations of motion (3.1.1), such contributions could arise only from a bulk complex being formed from two subnetwork proteins, which we have excluded. Similarly, as subnetwork complexes can only be formed from subnetwork proteins, $\boldsymbol{L}^{\mathrm{bb,s}}$ must vanish; in $\boldsymbol{L}^{\mathrm{sb,s}}$ only elements where the first and second index refer to the same subnetwork species can be nonzero, which then correspond to formation rates for bulk complexes from a subnetwork and a bulk protein.

For the equations of motion of quadratic observables, it is easiest to note that

$$\frac{\partial}{\partial t}\delta x_i \delta x_j = \frac{\partial \delta x_i}{\partial t}\delta x_j + \delta x_i \frac{\partial \delta x_j}{\partial t} = \sum_k \delta x_k \delta x_j L_{ki} + \sum_k \delta x_i \delta x_k L_{kj} + \delta x^3 + \mathcal{O}(\epsilon) \quad (3.2.43)$$

where the L_{ij} are the linear-linear entries of $\boldsymbol{L}$. The quadratic-quadratic blocks of $\boldsymbol{L}$, such as $L^{\mathrm{ss,ss}}$, can then be obtained directly from the linear-linear ones. Because all terms on the r.h.s. of (3.2.43) contain a factor of either δx_i or δx_j, one then sees that the blocks $\boldsymbol{L}^{\mathrm{ss,bb}}$ and $\boldsymbol{L}^{\mathrm{bb,ss}}$ are generically zero.

We summarize the discussion of the block structure of $\boldsymbol{L}$ briefly. Eq. (3.2.43) implies generically that all linear-quadratic blocks vanish as already shown in (3.2.33), and that $\boldsymbol{L}^{\mathrm{ss,bb}} = \boldsymbol{L}^{\mathrm{bb,ss}} = 0$. Our assumptions on the subnetwork-bulk division imply further that $\boldsymbol{L}^{\mathrm{ss,b}} = 0$ (bulk complex never formed from two subnetwork proteins) and $\boldsymbol{L}^{\mathrm{bb,s}} = 0$ (subnetwork complex always formed from two subnetwork proteins). Most entries of $\boldsymbol{L}^{\mathrm{sb,s}}$ are also zero except for those of the form $L_{(sb),s}$, where the same subnetwork species s appears in the quadratic first index and the linear second index. Here and in the following we use indices b, c, c' etc for bulk species and s, s', s'', u etc for subnetwork species to make the distinction obvious from the notation. These constraints then simplify the expression (3.2.41) for the nonlinear memory matrix considerably:

$$\boldsymbol{M}^{\mathrm{ss,s}}(\Delta t) = \boldsymbol{L}^{\mathrm{ss,sb}}[\boldsymbol{E}^{\mathrm{sb,b}}(\Delta t)\boldsymbol{L}^{\mathrm{b,s}} + \boldsymbol{E}^{\mathrm{sb,sb}}(\Delta t)\boldsymbol{L}^{\mathrm{sb,s}}] \quad (3.2.44)$$

Before looking at the consequences for the memory terms in the nonlinear projected equations of motion, we comment briefly on the local-in-time terms from the linear and nonlinear rate matrices, as shown in the first line of (3.2.18). The discussion above implies that reactions within the subnetwork contribute terms to the equations of motion for subnetwork concentrations only via $\boldsymbol{L}^{\mathrm{s,s}}$ and $\boldsymbol{L}^{\mathrm{ss,s}}$. As these just give the rate matrices, cf. (3.2.35) and (3.2.36), one deduces that all subnetwork reactions are captured, in their original form, in local-in-time terms. This was one of the desiderata for our projected description of the subnetwork dynamics.

More importantly, for the memory functions we can now deduce which can be nonzero in a given subnetwork. Let us term "boundary species" the molecular species from the subnetwork that interact directly with any bulk species, and "interior" species the others. Given our assumptions above, the interaction of a boundary species with the bulk could be a unary reaction, where a subnetwork species s is transformed into a bulk

species b (by phosphorylation, say). This would give nonzero entries in the blocks $\boldsymbol{L}^{\mathrm{b,s}}$ and $\boldsymbol{L}^{\mathrm{s,b}}$, specifically L_{bs} and L_{sb}, while such entries will be zero for interior species i. More commonly, a boundary subnetwork protein s and a bulk protein b can form a bulk complex c, contributing in addition to $\boldsymbol{L}^{\mathrm{sb,s}}$, via the element $L_{(sb),s}$, and to $\boldsymbol{L}^{\mathrm{sb,b}}$ via $L_{(sb),b}$ and $L_{(sb),c}$.

The first statement we can deduce about memory functions is that *memory terms appear only in the equations of motion for boundary species.* Mathematically, $M^{\mathrm{s,s}}_{s's}(t) = M^{\mathrm{ss,s}}_{(s's''),s}(t) = 0$ when s is an interior species. This follows directly from (3.2.40) and (3.2.44) because, from the discussion above, the s-th columns of $\boldsymbol{L}^{\mathrm{b,s}}$ and $\boldsymbol{L}^{*\mathrm{b,s}}$ vanish for an interior species s.

Turning now to boundary species s, we can further narrow down what memory functions can be nonzero. For the linear memory $M^{\mathrm{s,s}}_{s's}(t)$ to be nonzero, the index of the species s' influencing the evolution of s must be such that $L_{s'b}$ is nonzero for some bulk species b, giving a nonzero entry in the s'-th row of (3.2.40). As we saw above, this is possible only if s' is itself a boundary species, taking part in a unary or binary reaction with the bulk. The conclusion is that linear memory functions are nonzero *only for boundary species influencing other boundary species.*

There are similar restrictions on the entries of the nonlinear memory matrix $\boldsymbol{M}^{\mathrm{ss,s}}(t)$ for the time evolution of a boundary species s. Looking at (3.2.44), this matrix is proportional to $\boldsymbol{L}^{\mathrm{ss,sb}}$, so that $M^{\mathrm{ss,s}}_{(s's''),s}(t)$ can be nonzero only if there is a nonzero element of the matrix $\boldsymbol{L}$ of the form $L_{(s's''),(ub)}$ with b a bulk index. As s' and s'' are both in the subnetwork, then also u must be in the subnetwork (as $\boldsymbol{L}^{\mathrm{ss,bb}} = 0$). Our question then becomes: which subnetwork products $\delta x_{s'}\delta x_{s''}$ appear in the equation of motion for a subnetwork-bulk product $\delta x_u \delta x_b$? Looking at (3.2.43), one has

$$\partial_t(\delta x_u \delta x_b) = \delta x_u \partial_t \delta x_b + \ldots = \sum_{u'} L_{u'b}\delta x_u \delta x_{u'} + \ldots \tag{3.2.45}$$

where u' is a subnetwork index and the dots indicate terms that are irrelevant here because they do not involve the product of two subnetwork concentrations. One reads off that

$$L_{(s's''),(ub)} = \delta_{s'u}L_{s''b} + \delta_{s''u}L_{s'b} \tag{3.2.46}$$

when $s' < s''$, while for $s' = s''$

$$L_{(s's'),(ub)} = \delta_{s'u} L_{s'b} \tag{3.2.47}$$

Therefore one of s' and s'' must equal u, and the other index must be a reaction partner of the bulk species b, hence a boundary species. As u is an arbitrary subnetwork species, this means that *the only concentration products affecting the evolution of a boundary species via memory terms are products involving at least one boundary species.*

3.2.3.2 Memory function amplitudes

We next want to analyse the amplitudes of the memory functions at zero time difference, to see what they can tell us about the structure of the bulk and its interactions with the subnetwork.

Linear amplitudes: Self-memory. The linear memory matrix at zero time difference is, from (3.2.40), simply $\boldsymbol{L}^{\mathrm{s,b}}\boldsymbol{L}^{\mathrm{b,s}}$ because the exponential $\boldsymbol{E}^{\mathrm{b,b}}(\Delta t) = e^{\boldsymbol{L}^{\mathrm{b,b}}\Delta t}$ reduces to the identity matrix for $\Delta t = 0$. To calculate these amplitudes we can look at the possible structure of interactions between the subnetwork and bulk and then identify the terms in $\boldsymbol{L}^{\mathrm{s,b}}\boldsymbol{L}^{\mathrm{b,s}}$ that relate to these interactions. Initially we consider self memory functions $M^{\mathrm{s,s}}_{s,s}(0)$, which give the coefficient of $\delta x_s(t' = t)$ in the memory term of the equation for $\delta x_s(t)$. As discussed above, only boundary species in the subnetwork will have a nonzero self memory function.

Consider the self-memory for the subnetwork species s. By considering all possible reactions between subnetwork and bulk one finds for the relevant matrix elements of $\boldsymbol{L}$

$$\begin{aligned} L_{sb} &= \sum_c (k^+_{sc,b} y_c - k^+_{sb,c} y_b) + \lambda_{sb} \\ L_{bs} &= \sum_c (k^-_{b,sc} - k^+_{sb,c} y_s) + \lambda_{bs} \end{aligned} \tag{3.2.48}$$

and hence for the self-memory amplitude

$$M^{\mathrm{s,s}}_{ss}(0) = \sum_b L_{s,b} L_{b,s} = \sum_b \left[\sum_{c'} (k^+_{sc',b} y_{c'} - k^+_{sb,c'} y_b) + \lambda_{sb} \right] \left[\sum_c (k^-_{b,sc} - k^+_{sb,c} y_s) + \lambda_{bs} \right] \tag{3.2.49}$$

The first bracket here is a coefficient in the equation of motion for species b, so encodes the initial effect of a deviation δx_s of the concentration of s on the concentration δx_b,

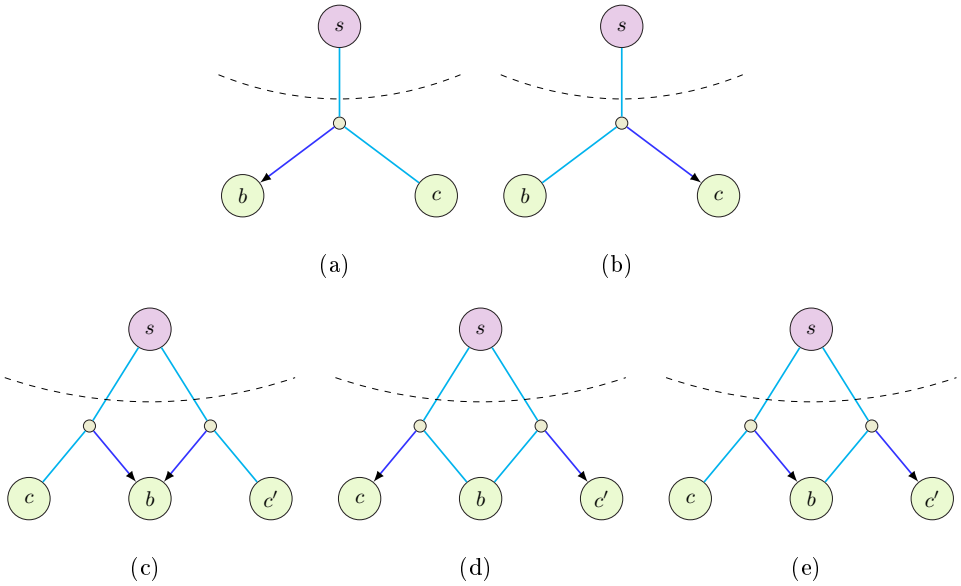


Figure 3.4: Sketch of interaction patterns for self-memory terms. The subnetwork species s reacts with (a) c in the bulk to make b; (b) b in the bulk to make c; (c) c and c' in the bulk to make b in the bulk; (d) b in the bulk to make c and c'; (e) c in the bulk to make b, which reacts again with s to produce c'.

while the second bracket gives the subsequent (after an infinitesimal time difference $t-t'$) feedback effect from b back to s. The different combinations of terms then correspond to different interaction patterns.

The product of the first terms in each bracket gives a contribution to the self-memory amplitude of

$$\sum_b \left(\sum_{c'} k^+_{sc',b} y_c \right) \left(\sum_c k^-_{b,sc} \right) \tag{3.2.50}$$

This is a contribution only from reactions where bulk species b is a complex formed from subnetwork protein s and another bulk protein c or c'. For the simplest case where there is only one such reaction involving b and s, this is sketched in Fig. 3.4(a). Intuitively, an increase in the concentration of s means that more b will be formed from the reaction with c. The bulk complex b will then dissociate again into s, thus increasing the rate of change of the concentration of s. This produces a positive self-memory amplitude.

The product of the second terms in each bracket in (3.2.49) also give a positive contri-

bution to the self-memory amplitude, but from a combination of two negative effects:

$$\sum_b \left(-\sum_{c'} k^+_{bs,c'} y_b \right) \left(-\sum_c k^+_{sc,l} y_s \right) \tag{3.2.51}$$

For the simplest case ($c = c'$) this is sketched in Fig. 3.4(b). Here the bulk species b that transmits the instantaneous memory forms a complex c' together with the subnetwork species s. An increase in the concentration of s then means that more of b will be used in the formation of c: the concentration of b is reduced. Consequently there will be less b to react with s, giving overall a positive effect on the rate of change of s.

If the subnetwork species s only takes part in one complex formation reaction with the bulk, then the two terms (3.2.50) and (3.2.51) give the total self-memory amplitude, which will be positive. The same is true if s reacts with the bulk in several ways but none of these reactions share bulk species.

If s is involved in several overlapping complex formation reactions with the bulk then one still gets the positive self-memory amplitude contributions (3.2.50) and (3.2.51), but now c and c' can be different as sketched in Figs. 3.4(c) and 3.4(d). In addition, however, there can be negative memory contributions where a positive initial effect from s on b combines with a negative subsequent effect, cf. Fig. 3.4(e), or vice versa. These are the cross terms from (3.2.49), reading explicitly:

$$\sum_b \left(\sum_{c'} k^+_{sc',b} y_{c'} \right) \left(-\sum_c k^+_{sb,c} y_s \right) + \sum_b \left(-\sum_{c'} k^+_{sb,c'} y_b \right) \left(\sum_c k^-_{b,sc} \right) \tag{3.2.52}$$

The first product accounts for the fact that an increase in s means that there will be more of it to react with c' to form b; b then reacts with s to make c, having a negative effect on the concentration of s. The second product corresponds to s reacting with b to form c' (negative effect) and then b dissociating into s and c (positive effect). Because such negative self-memory contributions rely on a single bulk species being both formed in one complex reaction and acting as reaction partner in a further complex formation, they necessarily involve ternary subnetwork-subnetwork-bulk complexes. On this basis one would expect that *positive* linear self-memory amplitudes are the norm and negative ones, where contributions of the type (3.2.52) would have to dominate, the exception.

We have not yet discussed the unary reaction contributions to the self-memory amplitude. Where such reactions do not overlap with other subnetwork-bulk reactions, they

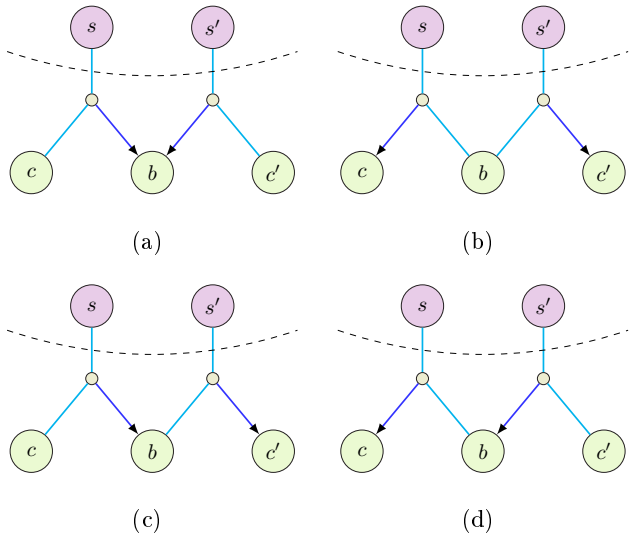


Figure 3.5: Interaction patterns for cross-memory amplitudes. See text for discussion.

make a positive contribution of $\lambda_{bs}\lambda_{sb}$ to the amplitude (3.2.49). Negative contributions would result only from overlap, where a unary reaction partner b of a subnetwork species s is also a reaction partner in a complex formation reaction with s.

Linear amplitudes: Cross-memory. All other linear memory function amplitudes $M^{\mathrm{s,s}}_{s's}(0)$, where the concentration of s' influences the rate of change of that of s, can be calculated similarly. Explicitly, each cross-memory amplitude is given by the analogue of (3.2.49),

$$M^{\mathrm{s,s}}_{s's}(0) = \sum_b L_{s'b}L_{bs} = \sum_b \left[\sum_{c'}(k^+_{s'c',b}y_{c'} - k^+_{s'b,c'}y_b) + \lambda_{s'b}\right]\left[\sum_c (k^-_{b,sc} - k^+_{sb,c}y_s) + \lambda_{bs}\right] \tag{3.2.53}$$

Leaving aside unary reactions, there are four possible cases as shown in Figure 3.5. A positive contribution to the amplitude is obtained when the subnetwork species s and s' have a shared product or a shared reactant b as in Figures 3.5(a) and 3.5(b), and a negative amplitude contribution is obtained when the shared species b is a reactant in one reaction and a product in the other reaction as in Figures 3.5(c) and 3.5(d). As before this relies on the existence of ternary subnetwork-subnetwork-bulk complexes and so positive contributions would generically be expected to dominate. For example in EGFR we only have two negative cross-memory amplitudes, in the cross terms between Grb2 and SOS as shown in Sec. 4.2.

Nonlinear amplitudes: Self-memory. We can look similarly at the amplitude of nonlinear memory functions $M^{\mathrm{ss,s}}_{(s's''),s}(0)$. These are the elements of the matrix $\boldsymbol{M}^{\mathrm{ss,s}}(0)$, which from (3.2.44) is given by $\boldsymbol{L}^{\mathrm{ss,sb}}\boldsymbol{L}^{\mathrm{sb,s}}$ because $\boldsymbol{E}(0)$ is the identity matrix. Recall now that the only nonzero elements of $\boldsymbol{L}^{\mathrm{sb,s}}$ are of the form $L_{(sb),s}$, so that

$$M^{\mathrm{ss,s}}_{(s's''),s}(0) = \sum_b L_{(s's''),(sb)} L_{(sb),s} \tag{3.2.54}$$

The first factor in turn was determined above in (3.2.46,3.2.47), while the second one is given explicitly by

$$L_{(sb),s} = -\sum_c k^+_{sb,c} \tag{3.2.55}$$

as can be read off from the equation of motion for δx_s.

We look at the simpler case $s' = s''$ first, which gives the quadratic self-memory amplitude. Inserting (3.2.47) into (3.2.54) and using the explicit form of $L_{(sb),s}$ and L_{sb} (cf. (3.2.48)) yields the constraint $s' = s$, so that the only nonzero amplitude for this case is

$$M^{\mathrm{ss,s}}_{(ss),s}(0) = \sum_b \left[\sum_{c'} (k^+_{sc',b} y_{c'} - k^+_{sb,c'} y_b) + \lambda_{sb}\right] \left(-\sum_c k^+_{sb,c}\right) \tag{3.2.56}$$

Pairing the first and second term in the first bracket with the second factor corresponds to the interaction patterns shown above in Figs. 3.4(e) and 3.4(d), respectively. The sign of each contribution to the quadratic self-memory amplitude is the same as the corresponding contribution to the linear self-memory. Explicitly, the second pairing from above gives the amplitude of the self product of 3.4(d). This is

$$\sum_b \left(-\sum_{c'} k^+_{sb,c'} y_b\right) \left(-\sum_c k^+_{sb,c}\right) \tag{3.2.57}$$

where an increase in s means that it will be used up in the reaction with b to form c'. There will then be less s to react with b to form c subsequently, having a positive effect on the rate of change of s. Note that such a contribution to the quadratic self-memory amplitude will be present for any boundary species, with the restriction $c = c'$ if it participates only in non-overlapping bulk reactions.

The first combination from (3.2.56) corresponds to Fig. 3.4(e) with c and c' swapped and gives a contribution

$$\sum_b \left(\sum_{c'} k^+_{sc',b} y_{c'}\right) \left(-\sum_c k^+_{sb,c}\right) \tag{3.2.58}$$

Here, an increase in s allows more b to be formed from s and c'. The b then reacts with s, having a negative effect on the rate of change of s.

Nonlinear amplitudes: Cross-memory. One can discuss nonlinear cross-memory amplitudes, where $s' < s''$, in a similar fashion. Starting from (3.2.46) one finds

$$M^{\mathrm{ss,s}}_{(s's''),s}(0) = \delta_{s''s} \sum_b \left[\sum_{c'} (k^+_{s'c',b} y_{c'} - k^+_{s'b,c'} y_b) + \lambda_{s'b} \right] \left(- \sum_c k^+_{sb,c} \right) + (s' \leftrightarrow s'') \tag{3.2.59}$$

The shorthand $(s' \leftrightarrow s'')$ indicates that the analogous term with s' and s'' swapped has to be added, to account for the two alternative cases $s' < s'' = s$ and $s = s' < s''$. The delta function prefactor indicates that we get nonzero amplitudes only for concentration products where one factor equals δx_s, the concentration of the boundary species being influenced; the other factor has to relate to a boundary species. More generic products, constrained only by the fact that one factor has to relate to a boundary species, can thus contribute to memory terms only at nonzero time difference.

As before one can combine each of the first two terms in the first factor in (3.2.59) with the second factor and then sees that these correspond to the interaction patterns sketched in Fig. 3.5(d) and 3.5(b), respectively. The signs of the amplitude contributions are again the same as for the linear cross-memory.

3.2.3.3 Memory function timescales

So far we have discussed the amplitude of the memory functions. Equally important for the overall effect of the memory terms is the timescale on which the memory functions decay. For a generic memory function $M(t)$ we will identify this timescale by dividing its time integral by the amplitude: $\tau = [1/M(0)] \int_0^\infty dt\, M(t)$. If $M(t)$ decays as a single exponential, $M(t) = M(0) e^{-t/\tau}$, this would give back the decay time τ. Alternatively we could define τ by asking when the function has decayed to some fraction of its initial value, say $M(t)/M(0) = 1/e$ for $t = \tau$. However, because $M(t)$ is in general a superposition of many exponentials, it is difficult to get closed form expressions for timescales with this definition.

Applying this definition to the linear self-memory $M^{\mathrm{s,s}}_{ss}(t)$ and using (3.2.40) gives an

explicit expression for the timescale of the form

$$\tau_s = \frac{[\boldsymbol{L}^{\mathrm{s,b}}(\boldsymbol{L}^{\mathrm{b,b}})^{-1}\boldsymbol{L}^{\mathrm{b,s}}]_{ss}}{[\boldsymbol{L}^{\mathrm{s,b}}\boldsymbol{L}^{\mathrm{b,s}}]_{ss}} \tag{3.2.60}$$

Qualitatively, one sees from this that each τ_s is a weighted average of elements of $(\boldsymbol{L}^{\mathrm{b,b}})^{-1}$. As the elements or $\boldsymbol{L}^{\mathrm{b,b}}$ are all proportional to reaction rates within the bulk, this shows that generally the memory function timescales will scale with the inverse of these rates: memory functions decay more quickly the faster the dynamics in the bulk. One can check this explicitly by scaling up all bulk reaction rates by a certain factor; the timescales τ_s will then decrease by the same factor. One proviso here is that the steady state concentrations y_i must be maintained because contributions of $\boldsymbol{L}^{\mathrm{b,b}}$ from complex formation reactions are of the form $k^+_{\cdot}y$. In the example (3.1.14), one would need to keep the ratio of $k^+_{14,5}$ and $k^-_{5,14}$ constant while scaling up both rates.

In practice the reaction rates have to be treated as given so the scaling argument does not necessarily help to understand the order of magnitude of the memory timescales τ_s. One concept that can provide quantitative insight is to think of the memory functions as decomposed according to source and receiver "channels". The (linear) memory matrix (3.2.40) is proportional to $\boldsymbol{L}^{\mathrm{s,b}}$ and $\boldsymbol{L}^{\mathrm{b,s}}$. Both of these can be seen as a superposition of contributions from individual reactions between boundary species and bulk. If we identify each such contribution as a source channel (for $\boldsymbol{L}^{\mathrm{s,b}}$) or a receiver channel (for $\boldsymbol{L}^{\mathrm{b,s}}$), then the memory function is a sum of all possible contributions of source and receiver channels. Here the source tells us via which boundary reaction a concentration fluctuation in the past initially propagated into the bulk, and the receiver channel defines by which route it feeds back into the subnetwork. We will explore this decomposition of memory functions in the concrete example that is the subject of the next section.

Each combination of source and receiver will have a specific "propagation timescale" in the bulk, which will consist of combinations of entries of $(\boldsymbol{L}^{\mathrm{b,b}})^{-1}$. The overall memory timescale τ_s from (3.2.60) can then be viewed as a weighted average of these propagation timescales, but bearing in mind that the weights can be both positive and negative.

The memory functions are dominated by the reactions that have larger contributions in the blocks $\boldsymbol{L}^{\mathrm{s,b}}$ and $\boldsymbol{L}^{\mathrm{b,s}}$. For example, if we consider a reaction of the form $s+b \to b'$ then this will dominate if either the steady state concentration y_b or the complex formation rate $k^+_{sb,b'}$ are large enough for the product $k^+_{sb,b'}y_b$ to be significantly larger than for any

competing reaction.

A very similar decomposition can be performed for nonlinear memory functions, using (3.2.44). If we write $\boldsymbol{QLQ}^{-1}$ in condensed block form as

$$\boldsymbol{QLQ}^{-1} = \begin{pmatrix} (\boldsymbol{L}^{\mathrm{b,b}})^{-1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{I}^{\mathrm{sb,b}} & \boldsymbol{I}^{\mathrm{sb,sb}} & \boldsymbol{I}^{\mathrm{sb,bb}} \\ \boldsymbol{I}^{\mathrm{bb,b}} & \boldsymbol{I}^{\mathrm{bb,sb}} & \boldsymbol{I}^{\mathrm{bb,bb}} \end{pmatrix} \tag{3.2.61}$$

then the timescale for e.g. the nonlinear self-memory functions is

$$\tau_{ss} = \frac{\left[\boldsymbol{L}^{\mathrm{ss,sb}}\left(\boldsymbol{I}^{\mathrm{sb,b}}\boldsymbol{L}^{\mathrm{b,s}} + \boldsymbol{I}^{\mathrm{sb,sb}}\boldsymbol{L}^{\mathrm{sb,s}}\right)\right]_{(ss),s}}{\left[\boldsymbol{L}^{\mathrm{ss,sb}}\boldsymbol{L}^{\mathrm{sb,s}}\right]_{(ss),s}} \tag{3.2.62}$$

The source channels are identified by grouping the contributions to $\boldsymbol{L}^{\mathrm{ss,sb}}$, which are given explicitly in (3.2.46) and (3.2.47), according to reactions between boundary and bulk species. The terms in $\boldsymbol{L}^{\mathrm{b,s}}$ and $\boldsymbol{L}^{\mathrm{sb,s}}$, cf. (3.2.55), can be grouped analogously into receiver channels.

3.3 Summary

In this Chapter I have shown how to apply the projection method to protein interaction networks. In Section 3.2, I showed how to calculate the memory functions for both linearised and nonlinear dynamics. I then looked at different properties of memory functions (Sec. 3.2.3). We find that the only species with memory functions are boundary species that interact directly with the bulk. Analysing the timescales and amplitudes of the memory functions provides insights on how the subnetwork interacts with the bulk.

The full projected equations are calculated in the limit of $\epsilon \to 0$ and therefore define a subnetwork based on the rate equations only. The projection approach can also capture noise effects via the random force terms. This noise is the component of the extrinsic noise that comes from the bulk initial conditions, not the more general extrinsic noise as defined in, for example, [47].

Chapter 4

Application to EGFR

I now apply the projection method to a model for the signalling network of epidermal growth factor receptor (EGFR) developed by Kholodenko et al. [13], see Figure 4.1. It is a relatively small and well-studied network [48, 49] and contains a number of subnetworks, such as Src homology and collagen domain protein (Shc) and Shc-interacting proteins. The EGFR network is part of the family of protein-tyrosine kinase receptors that regulate cell growth and differentiation and is relevant to cancer signalling [50].

4.1 Setup of EGFR model for application of projection technique

We use the mass action rate parameters from [13]. Most of these fit directly into our setup of unary and binary reactions. The network also involves three Michaelis-Menten reactions transforming a "substrate" into a "product". We incorporate these by adding to the description one enzyme and one enzyme-substrate complex species per reaction, with the rates for formation and dissociation of the complex large enough in order to force the two added species to be in equilibrium at any time with the prevailing substrate and product concentrations [51]. Initial conditions for the added species are then also derived from those of the substrate according to this equilibrium criterion. We choose the above method of incorporating Michaelis-Menten reactions as it allows direct application of the framework developed so far. We report in Chapter 5 on an alternative approach

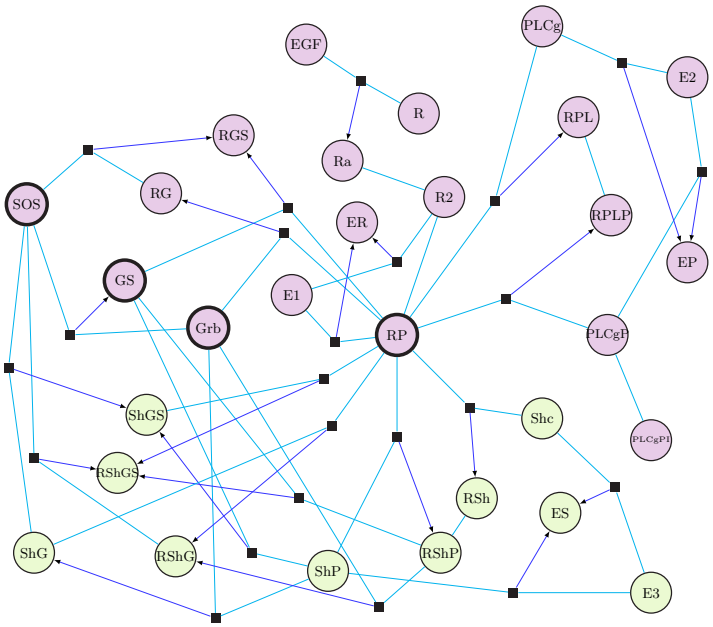


Figure 4.1: Factor graph of EGFR network as described in Kholodenko et al. [13] adapted to include enzyme reactions. Three added enzyme reactions with enzymes denoted E1-3 and enzyme-substrate complexes "enzyme-R" (denoted ER), "enzyme-PLCγ" (denoted EP), "enzyme-Shc" (denoted ES) are also shown to capture Michaelis-Menten contributions to the dynamics. The purple nodes denote subnetwork species while the bulk species are shown in green. See Appendix A for a full list of abbreviations for each network component.

where the enzyme reactions do not need to be represented explicitly.

To apply the projection method we need to first select a subnetwork and bulk from the EGFR network. We have chosen the bulk to be the protein Src homology and collagen domain protein (Shc) and any complexes that include Shc, consistent with our convention that if a protein is in the bulk, any complexes containing that protein will also be in the bulk. Shc and its complexes interact directly with four subnetwork species, which therefore form the boundary of our subnetwork; they are phosphorylated EGFR (denoted RP), growth factor receptor-binding protein 2 (Grb2), Son of Sevenless homolog protein (SOS), and protein complex Grb2-SOS (denoted GS; see Appendix A for a full list of abbreviations for network components). We then apply the projection method to obtain a set of equations for the subnetwork species. For the interior species these will have the original mass action form, while the boundary species will acquire additional memory (and random force) terms.

To avoid having to carry around concentration units in the following, we will switch to dimensionless concentrations defined as

$$\delta\tilde{x}_i = \delta x_i / y_i \tag{4.1.1}$$

The $\delta\tilde{x}_i$ are dimensionless because the steady state concentration y_i and the difference δx_i between the actual concentration and the steady state value both have dimensions of concentration. Intuitively, the $\delta\tilde{x}_i$ are fractional concentration deviations from the steady state. The lowest value is -1, corresponding to a concentration of zero (100% below steady state), while e.g. $\delta\tilde{x}_i = 2$ indicates that the concentration of i is three times that in steady state.

The projected equations (3.1.7) written in terms of the dimensionless $\delta\tilde{x}_i$ take the form

$$\begin{aligned}\frac{\partial}{\partial t}\delta\tilde{x}_i = &\sum_{j=1}^{N^{\mathrm{s}}} \delta\tilde{x}_j \tilde{\Omega}^{\mathrm{s,s}}_{ji} + \sum_{1\le j\le k\le N^{\mathrm{s}}} \delta\tilde{x}_j \delta\tilde{x}_k \tilde{\Omega}^{\mathrm{ss,s}}_{(jk)i} \\ &+ \int_0^t dt' \left(\sum_{j=1}^{N^{\mathrm{s}}} \delta\tilde{x}_j(t') \tilde{M}^{\mathrm{s,s}}_{ji}(t-t') + \sum_{1\le j\le k\le N^{\mathrm{s}}} \delta\tilde{x}_j(t')\delta\tilde{x}_k(t') \tilde{M}^{\mathrm{ss,s}}_{(jk)i}(t-t') \right) + \tilde{r}_i(t)\end{aligned} \tag{4.1.2}$$

where

$$\begin{gathered}\tilde{\Omega}^{\mathrm{s,s}}_{ji} = y_j \Omega^{\mathrm{s,s}}_{ji} y_i^{-1}, \qquad \tilde{\Omega}^{\mathrm{ss,s}}_{(jk)i} = y_j y_k \Omega^{\mathrm{ss,s}}_{(jk)i} y_i^{-1}, \\ \tilde{M}^{\mathrm{s,s}}_{ji}(\Delta t) = y_j M^{\mathrm{s,s}}_{ji}(\Delta t) y_i^{-1}, \qquad \tilde{M}^{\mathrm{ss,s}}_{(jk)i}(\Delta t) = y_j y_k M^{\mathrm{ss,s}}_{(jk)i}(\Delta t) y_i^{-1}, \qquad \tilde{r}_i(t) = r_i(t) y_i^{-1}\end{gathered} \tag{4.1.3}$$

The rescaled rate matrix entries and random forces have dimensions of rate, i.e. inverse time, while the rescaled memory functions have dimensions of rate squared. The rate matrices and memory functions are calculated by first constructing the matrix $\boldsymbol{L}$ for the network, then using (3.2.35), (3.2.36), (3.2.40) and (3.2.44), and finally switching to dimensionless concentrations as explained above.

We will first discuss qualitative features of the memory functions themselves. Quantitative tests of the projected equations are presented next; as before, we will drop the random force terms so that the equations are a closed system determining the time-courses of the subnetwork concentrations for any given initial condition. To solve this numerically, we implement a solver for systems of integro-differential equations [52]. The enzyme reactions are much faster than the remainder of the kinetics and this causes the

system of equations to become what is known as "stiff". This can be handled by converting enzyme reaction terms in the subnetwork back into Michaelis-Menten form before using the numerical solver, or by transforming the projected equations into an enlarged set of differential equations (see Appendix B) that can then be integrated using standard methods for stiff systems [53].

4.2 Memory function properties

We will now look at how general properties of the memory functions described in Section 3.2.3 manifest themselves in the EGFR network. We first analyse the amplitudes of the memory functions as in Section 3.2.3.2, to see how these reflect the structure of the network.

Figure 4.2 shows two of the linear memory functions in the equation of motion for the concentration of Grb2, one the self-memory and the other the cross-memory that determines the influence of past concentration values of SOS. The amplitudes are given by the intercepts with the y-axis ($\Delta t = 0$): one sees that the self-amplitude of Grb2 is positive. To understand why, we note that Grb2 has two different reactions with bulk species:

$$\begin{aligned} \text{Grb2} + \text{ShP} &\leftrightharpoons \text{ShG} \\ \text{Grb2} + \text{RShP} &\leftrightharpoons \text{RShG} \end{aligned} \tag{4.2.1}$$

As these do not overlap, each reaction gives separate contributions to the self-memory amplitude of the form shown in Figs. 3.4(a) and 3.4(b), which are always positive.

The amplitude of the cross-memory of $\delta\tilde{x}_{\text{GRB}}$ to $\delta\tilde{x}_{\text{SOS}}$, on the other hand, is negative as Figure 4.2 shows. To rationalise this, note that there are two bulk species that are shared between the bulk reactions of SOS and Grb2, namely ShG (ShP-Grb2) and RShG (RP-Shc-Grb2). The reaction patterns involving these species both have the structure of Fig. 3.5(c), and hence both give negative contributions.

Next we look at the time-dependence of the memory functions, and in particular the channel decomposition described in Sec. 3.2.3.3. For the self-memory of Grb2, there are two source and receiver channels, namely the two bulk reactions (4.2.1). The memory function can be decomposed into four pieces according to the combination of these four

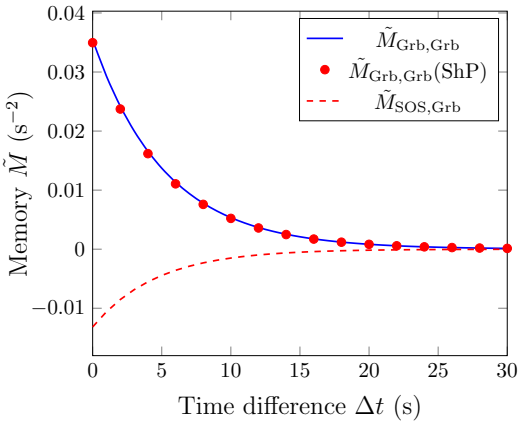


Figure 4.2: Memory functions in EGFR network: coefficients of Grb2 and SOS in the linear memory term for Grb2. The self memory function of Grb2 is compared to the contribution coming only from the reaction with phosphorylated Shc (ShP) as source and receiver channel. Contributions from other channels are no larger than $8 \cdot 10^{-4}\mathrm{s}^{-2}$.

channels, e.g. "out via ShP (source) and in via RShP (receiver)". It turns out in this case that the channel via phosphorylated Shc (ShP) dominates entirely. This is shown in Fig. 4.2, which compares the total memory function with its "out and in via ShP" contribution.

Note that the rates for both reactions (4.2.1) between Grb2 and the bulk are the same; however the steady state concentration of phosphorylated Shc (ShP) is much larger than the steady state value of RShP and therefore the reaction between Grb2 and phosphorylated Shc (ShP) to make ShP-Grb2 (ShG) is the one that dominates the self memory function of Grb2. Therefore it may be useful to study interactions between Shc and Grb2 to help understand why this reaction dominates the memory.

For the self-memory of phosphorylated EGFR (RP), which has four interactions with the bulk, the channel decomposition is richer because there are now 16 combinations of the four source and receiver channels. None of the $k^+_{sb,c}y_b$ values for the interactions with the bulk proteins is large enough to be entirely dominant and accordingly there are several channel combinations that give significant contributions to the memory. Fig. 4.3 shows the four that are largest: in and out via Shc; in and out via ShP; and in via Shc and out via ShP and vice versa. The combination of these gives a good account of the overall shape of the memory function, indicating that two channels (Shc and

phosphorylated Shc (ShP)) are dominant over the other two (ShP-Grb2 (ShG) and Shc-Grb2-SOS (ShGS)). Looking at the figure closely one sees that the cross-channel contribution between Shc and ShP is positive for short time differences but becomes negative (and small) for longer time differences. In contrast the two other terms, for which the source and receiver channel is the same, are always positive.

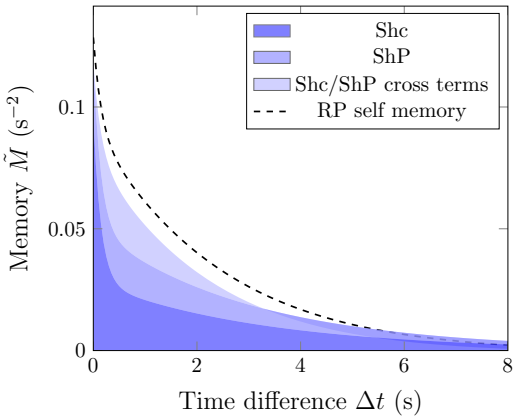


Figure 4.3: Comparison of self memory function of phosphorylated EGFR (RP) with dominant terms from the channel decomposition: in and out via Shc; in and out via phosphorylated Shc (ShP); and in via Shc and out via phosphorylated Shc (ShP) and vice versa, shown together.

An analogous channel decomposition can be performed for (linear) cross memory and nonlinear memory functions as explained in Sec. 3.2.3.3. As for the linear self memory functions above, we find that often only a few channels provide the dominant contribution. This occurs for all the memory functions of EGFR. Fig. 4.4 shows that the nonlinear self memory of Grb2 is dominated by the reaction Grb2 + ShP $\rightarrow$ ShG acting as source and receiver, i.e. by the channel combination "in and out via phosphorylated Shc (ShP)". This is not unexpected as the same combination dominates the linear self memory (see Fig. 4.2).

The channel decomposition can also be used to analyse self memory function timescales τ_s as defined in (3.2.60). In particular, if there is a single channel that dominates the memory function then the memory function contribution from this channel will have a similar timescale to the full memory function. For example, we find that the self memory of Grb2 has a timescale $\tau_{\mathrm{Grb}} = 5.31\,\mathrm{s}$. The contribution from the phosphorylated Shc (ShP) channel as source and receiver, shown in Fig. 4.2, has a timescale that is very

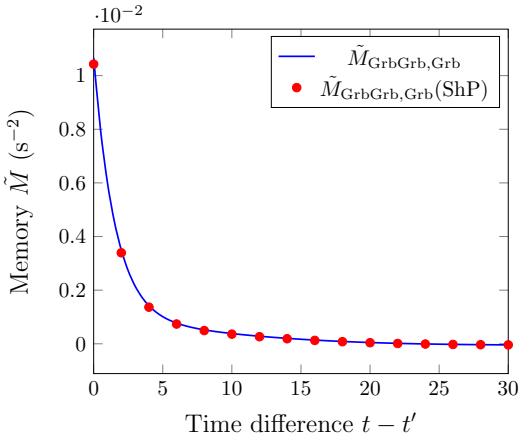


Figure 4.4: Comparison of nonlinear self memory function of Grb2 with the contribution from the channel combination "in and out via phosphorylated Shc (ShP)".

close to this, namely 5.25 s.

The memory function of phosphorylated EGFR (RP) has a faster timescale, $\tau_{\mathrm{RP}} = 1.68\,\mathrm{s}$. The contributions to the memory function which come from the "in and out via Shc" and "in and out via phosphorylated Shc (ShP)" channel combinations, on the other hand, have timescales of 1.08 s and 2.86 s, respectively. The timescales of the contributions from the dominant channels are therefore sufficient to give an order-of-magnitude estimate of the overall memory function timescale.

The dominance of certain channels encourages us to look at how the system behaves if species or reactions that do not appear to contribute to the behaviour of the system are removed. The reaction between Grb2 and RShP does not make a significant contribution to the memory functions of Grb2. Excluding this reaction does not cause many changes to most of the memory functions involving Grb2, but some memory functions including $\tilde{M}_{\mathrm{RP,Grb}}$ have large differences. The change in $\tilde{M}_{\mathrm{RP,Grb}}$ occurs because removing the reaction between Grb2 and RShP means that phosphorylated EGFR (RP) and Grb2 only share interactions through phosphorylated Shc (ShP) and ShP-Grb2 (ShG). On the other hand, the channel decomposition shows that "in via Shc and out via RP-Shc-Grb2 (RShG)" and "in via phosphorylated Shc (ShP) and out via RP-Shc-Grb2 (RShG)" are the dominant reactions in this memory function and therefore removing this connection between phosphorylated EGFR (RP) and Grb2 will have a large effect on the memory behaviour. Therefore because all the channels are connected one cannot

necessarily remove channels that look weak in one memory function, as this will generally have an effect on the other species.

One benefit of our analysis is that we can characterise explicitly also the nonlinear memory functions, and in particular assess the relative size of their contribution compared to the linear memory terms. Figure 4.5 shows the nonlinear self memory functions of the boundary species and Table 4.1 lists the amplitudes and timescales of the linear and nonlinear self memory functions. It is easy to see that the nonlinear self memory amplitudes are all smaller than their respective linear amplitudes. Similarly the nonlinear self memory functions decay faster than their respective linear contributions as shown by their shorter timescales. This suggests that, where it is desirable to capture nonlinear memory terms only approximately, relatively simple approximations like short-timescale exponentials could be considered. The nonlinear self memory of RP is a special case: the memory function changes sign (see Fig. 4.5) and the positive and negative contributions to the integral defining the timescale (see Sec. 3.2.3.3) cancel almost exactly, giving a notional timescale that is much shorter than for the other boundary species.

There are at least two ways one could use information from the memory functions to estimate the values of the dimensionless concentrations where nonlinearities become important. Concentrating on the self memory functions as above, the linear and nonlinear instantaneous (small Δt) contributions become comparable when $\tilde{M}^{\mathrm{s,s}}(0)\delta\tilde{x}_s = \tilde{M}^{\mathrm{ss,s}}(0)\delta\tilde{x}_s^2$, leading to the estimate $\delta\tilde{x}_s^{\mathrm{c},1} = \tilde{M}^{\mathrm{s,s}}(0)/\tilde{M}^{\mathrm{ss,s}}(0)$ for the size of the linear regime. More relevant for the long-time dynamics is to consider the total memory terms assuming constant $\delta\tilde{x}_s(t)$. Then $\delta\tilde{x}_s^{\mathrm{c},2} = \int_0^\infty dt\, \tilde{M}^{\mathrm{s,s}}(t)/\int_0^\infty dt\, \tilde{M}^{\mathrm{ss,s}}(t)$ would delimit the extent of the linear regime, i.e. the ratio of amplitude times timescale for the linear and nonlinear self memory.

The two estimates $\delta\tilde{x}_s^{\mathrm{c},1}$ and $\delta\tilde{x}_s^{\mathrm{c},2}$ defined above are shown in Table 4.1 alongside the memory amplitudes and timescales. We see that e.g. for Grb2-SOS (GS) $\delta\tilde{x}^{\mathrm{c},2} = 29.92$ whereas for Grb2 $\delta\tilde{x}^{\mathrm{c},2} = 9.37$; this suggests that for Grb2 nonlinear memory functions have a larger effect. To confirm this, we have run numerical experiments on time courses starting in steady state except for a perturbation in one of the four boundary species (see Table 4.1). We then compared the time courses for this species as predicted from the projected equations with and without the nonlinear memory terms, respectively,

measuring the deviation between them as in Eq. (4.4.1) below. We find that these deviations are ordered among the four species in inverse proportion to their $\delta\tilde{x}^{\mathrm{c},2}$, i.e. the larger this measure of the size of the linear regime, the smaller the nonlinear memory effects. Among the three species other than phosphorylated EGFR (RP) we also find quantitatively that deviations observed for an initial perturbation $\delta\tilde{x}_s(0)$ chosen as some constant fraction (say 1%) of $\delta\tilde{x}_s^{\mathrm{c},2}$ are of similar magnitude. This indicates that values of $\delta\tilde{x}^{\mathrm{c},2}$ can give not just qualitative but also quantitative information. Our tests show it to be superior to $\delta\tilde{x}^{\mathrm{c},1}$ in this regard. For phosphorylated EGFR (RP), where because of the small notional nonlinear self memory timescale the value of $\delta\tilde{x}_s^{\mathrm{c},2}$ is unrealistically large, we find that it still retains qualitative significance: the deviations that we measure due to the omission of the nonlinear memory terms are the smallest (by a factor of 100 compared to the next largest) among the four species tested.

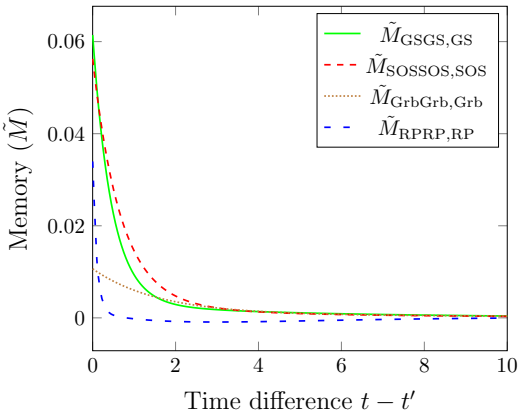


Figure 4.5: The nonlinear self memory functions of Grb2-SOS (GS), SOS, Grb2 and phosphorylated EGFR (RP).

4.3 Changing the subnetwork

If we take the EGFR network and choose a different subnetwork then we will have a different set of boundary nodes and memory functions. Although the memory functions will be quantitatively different, their behaviour will still adhere to the general principles derived above. Let us take the EGFR network and change the subnetwork so that the bulk consists of Grb2 and all complexes that include Grb2. The boundary species of the subnetwork are now phosphorylated EGFR (RP), SOS, RShP and phosphorylated Shc

Boundary species	linear		nonlinear		$\delta\tilde{x}_s^{c,1}$	$\delta\tilde{x}_s^{c,2}$
	amp.	τ	amp.	τ		
RP	0.13	1.68	0.034	$-2.16 \cdot 10^{-6}$	3.82	$2.97{\cdot}10^6$
Grb2	0.036	5.31	0.01	2.04	3.6	9.37
SOS	0.12	4.61	0.056	0.82	2.14	12.05
GS	0.23	5.23	0.06	0.67	3.83	29.92

Table 4.1: The amplitudes (in s^{-2}) and timescales (in s) of linear and nonlinear self memory functions. Also shown are the estimates for the size of the linear regime resulting from instantaneous and long-time memory contributions, $\delta\tilde{x}_s^{c,1}$ and $\delta\tilde{x}_s^{c,2}$, respectively.

(ShP). Figure 4.6 shows the linear self memory function of phosphorylated Shc (ShP) in the equation for ShP. ShP has two reactions with bulk species

$$\begin{aligned} \text{ShP} + \text{Grb2} &\leftrightharpoons \text{ShG} \\ \text{ShP} + \text{GS} &\leftrightharpoons \text{ShGS} \end{aligned} \tag{4.3.1}$$

As with the self memory function of Grb2 in Sec. 4.2 each reaction gives a separate positive contribution to the self-memory amplitude.

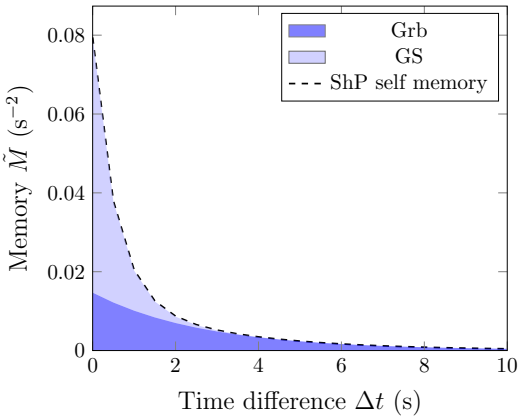


Figure 4.6: Comparison of self memory function of phosphorylated Shc (ShP) with dominant terms from the channel decomposition: in and out via Grb2; and in and out via Grb2-SOS (GS), shown together.

Next we consider the cross-species effects in the memory term for the evolution of ShP. The amplitude of the memory to past values of $\delta\tilde{x}_{\text{SOS}}$ is negative, whereas the amplitudes

for memory to $\delta\tilde{x}_{\mathrm{RP}}$ and $\delta\tilde{x}_{\mathrm{RShP}}$ is positive, as shown in Fig. 4.7. This last amplitude is made up of two reaction patterns with the structure of Fig. 3.5(b) and hence has to be positive as we find. The cross memory functions of SOS and phosphorylated EGFR (RP) consist of a mixture of different reaction patterns from Fig. 3.5, and the sign of their amplitude is therefore determined by the relative sizes of the contributions of different signs. The amplitude of the cross memory function of SOS ($\tilde{M}_{\mathrm{SOS,ShP}}$) is negative, whereas the amplitude of the cross memory function of phosphorylated EGFR (RP) ($\tilde{M}_{\mathrm{RP,ShP}}$) is positive but approximately ten times larger in size than the amplitude of the memory function for SOS.

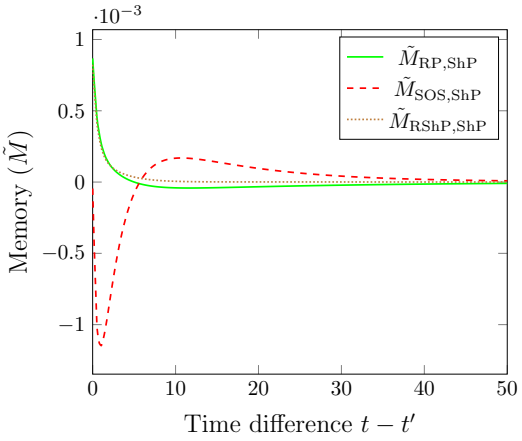


Figure 4.7: Memory functions in EGFR network: coefficients of phosphorylated EGFR (RP), SOS and RShP in the linear memory term for phosphorylated Shc (ShP).

Also for the current changed subnetwork one can decompose the self memory functions into channels to understand the relative important of the latter. Fig. 4.6 shows that the two channels "out and in via Grb2" and "out and in via Grb2-SOS (GS)" both give similar contributions to the memory function of ShP. This is in line with the earlier analysis in Sec. 4.2: the $k^{+}_{sb,c}y_b$ values for both interactions are of the same order and therefore one would expect that neither channel will dominate the other.

Looking finally at the memory function timescales, the self memory function of phosphorylated Shc (ShP) has a timescale of $\tau_{\mathrm{ShP}} = 0.97$s. The "in and out via Grb2" and "in and out via Grb2-SOS (GS)" channels have timescales of 2.84s and 0.54s, respectively, and one sees that these can again be used to give an order of magnitude estimate of the full memory function timescale.

4.4 Quantitative tests

We conclude our discussion of the EGFR network by analysing the quantitative accuracy of the projected equations. As before we focus on the limit of low copy number noise ($\epsilon \to 0$) and drop the random force terms to have a closed description of the subnetwork dynamics. Our baseline is the solution of the full set of reaction equations for the entire network, consisting of both subnetwork and bulk. We compare the performance of the projected equations, including memory terms, to two simpler approximations without memory. In the first one we treat the subnetwork as isolated, i.e. all subnetwork-bulk reactions are ignored. In the second one we assume the bulk dynamics is fast enough for the bulk to be in steady state with respect to the specific subnetwork concentrations at any given time [37]. In practice, this means we solve the steady state conditions for the bulk concentrations at every time step and substitute them into the evolution equations for the subnetwork. All three approximation methods (projected equations without random force, isolated subnetwork, steady state bulk) come in two versions, one derived from the linearised dynamics and one for the full nonlinear dynamics.

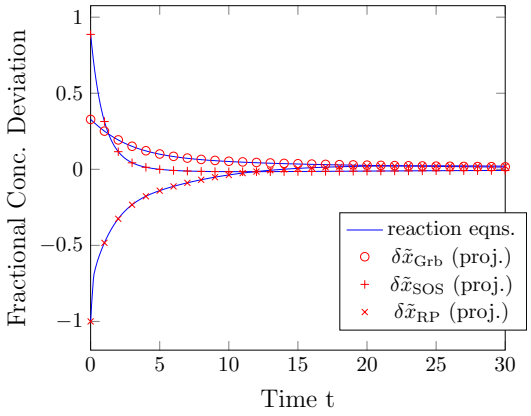


Figure 4.8: Plots of time courses of some selected molecular species from the EGFR network. The fractional concentration deviations (4.1.1) are defined so that 0 represents steady state concentration, which is approached for long time, and -1 represents zero concentration. The solutions to the nonlinear projected equations are visually indistinguishable from those of the full reaction equations. Initial conditions were chosen as explained in the text.

Figure 4.8 compares the solutions of the nonlinear projected equations to the baseline, the nonlinear reaction equations for the entire network. Time courses for phosphorylated

EGFR (RP), Grb2 and SOS are shown. Here and in the following, the subnetwork initial conditions were chosen to maximise nonlinear effects: specifically we maximised $\sum_s [\delta\tilde{x}_s(0)]^2$ subject to the constraint that all conserved concentrations have the same value as at the steady state given by $\delta\tilde{x}_i = 0$, and of course that all concentrations are non-negative ($\delta\tilde{x}_s(0) \geq -1$). The bulk was assumed to be in steady state initially, i.e. we set $\delta\tilde{x}_b(0)$ for all bulk species. As the figure shows, the agreement between the nonlinear projected equations and the full dynamics is excellent, with the two sets of time courses being visually indistinguishable.

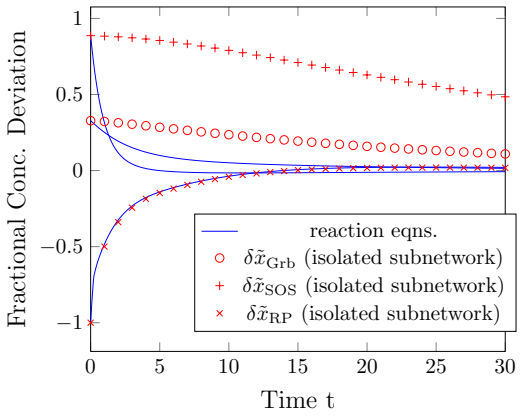


Figure 4.9: Comparison of time courses obtained from full reaction equations with time courses for isolated subnetwork. Initial conditions chosen as in Fig. 4.8.

To demonstrate the importance of accounting for the interactions of the subnetwork with the bulk, we contrast in Fig. 4.9 the solutions for the isolated subnetwork to those of the full reaction equations: substantial differences appear, with the relaxation to steady state predicted to occur over a much larger timescale than in the full description.

The approximation of retaining information on the bulk network but assuming the bulk dynamics is fast would be expected to provide a more accurate description. This is borne out by Fig. 4.10, though deviations from the baseline are still larger than for the nonlinear projected equations, emphasizing the importance of keeping track of memory effects.

To develop a more quantitative picture of the performance of the various approximations for the subnetwork dynamics, we consider the same initial conditions as above but now scale down all the $\delta\tilde{x}_s(0)$ by a constant factor to tune the initial deviation of the subnetwork from steady state. The magnitude of this deviation will be quantified via

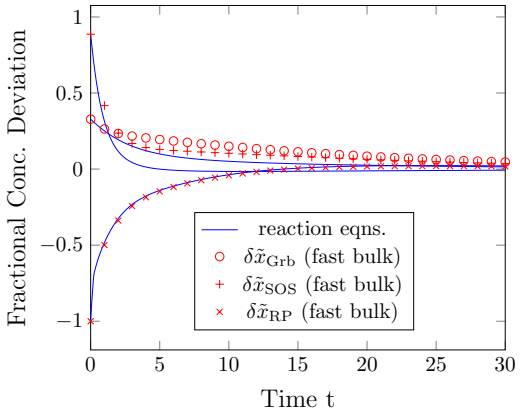


Figure 4.10: Comparison of time courses obtained from full reaction equations with time courses found by assuming the bulk dynamics is fast enough for the bulk to be at steady state. Initial conditions chosen as in Fig. 4.8.

the initial root mean squared deviation, $\delta = \left(\{\sum_s [\delta\tilde{x}_s(0)]^2\}/N^{\mathrm{s}}\right)^{1/2}$, where N^{s} is the number of subnetwork species as before. The accuracy of any approximation $\delta\hat{x}_s(t)$ for the subnetwork time courses will be measured by

$$\Delta = \frac{1}{T}\int_0^T dt' \frac{1}{N^{\mathrm{s}}} \sum_{s=1}^{N^{\mathrm{s}}} |\delta\tilde{x}_s(t) - \delta\hat{x}_s(t)| \tag{4.4.1}$$

This is the absolute deviation in the dimensionless concentration of each subnetwork species, averaged over species and also a time interval T that we choose as $T = 150$s to capture the interesting transient regime, i.e. the approach to the steady state.

If we now consider first the linear projected equations and compare them to the full nonlinear reaction equations, we would expect the error to increase quadratically with the size δ of the initial deviations from steady state, at least for small δ, because we are missing the nonlinear terms but are correctly capturing all linear terms including the memory. Fig. 4.11 verifies this expectation, showing that the average deviation Δ grows only as δ^2. By contrast, the simpler approximations derived from the linearised dynamics, where we treat the subnetwork as isolated or the bulk as fast, should show deviations from the true time courses already at order δ because they neglect linear memory terms. Fig. 4.11 is consistent with this. It demonstrates in addition that the errors made by the memoryless approximations are substantially larger in absolute terms than for the projected equations. This demonstrates that memory terms are essential even to describe the linearised dynamics correctly.

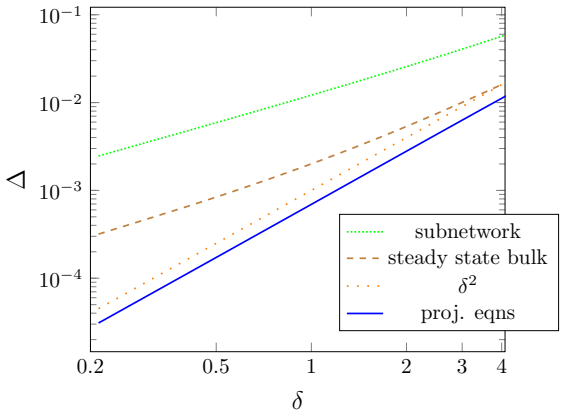


Figure 4.11: Plot of approximation error vs initial deviation from steady state in log-log representation, for three approximation methods derived from the linearised dynamics: linear projected equations, i.e. incorporating memory terms; steady state bulk approximation, i.e. without memory; and isolated subnetwork approximation. Dotted line is proportional to δ^2 to demonstrate that the approximation error of the linear projected equations is only quadratic in δ.

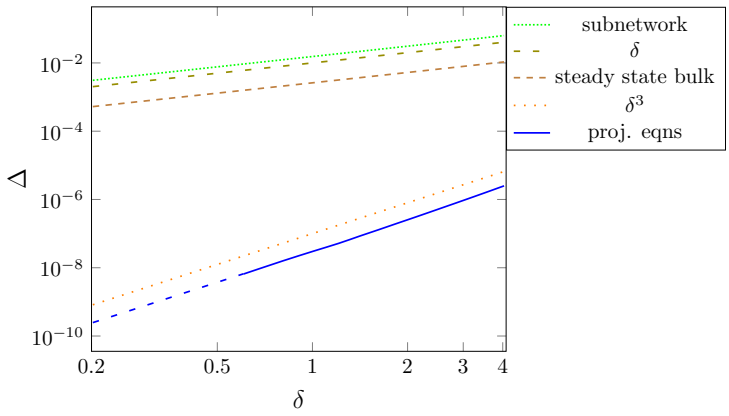


Figure 4.12: Plot of approximation error vs initial deviation from steady state in log-log representation, for three approximation methods derived from the nonlinear dynamics: nonlinear projected equations, i.e. incorporating memory terms; steady state bulk approximation, i.e. without memory; and isolated subnetwork approximation. Dotted line is proportional to δ^3 to demonstrate that the approximation error of the nonlinear projected equations grows only cubically in δ. The dashed blue line indicates the estimated size of the error in the projected equations in the region where it is too small for the given initial conditions to calculate accurately.

In Fig. 4.12 we show an analogous comparison for the nonlinear approximation methods. The memoryless approximations, isolated subnetwork and steady state bulk, still

fail to catch memory contributions that are present already in the linearised dynamics, and accordingly give an approximation error that grows linearly in δ. For the nonlinear projected equations, on the other hand, the approximation error comes only from the neglected random force terms. As explained in Section 3.2.2, when the bulk is initially in steady state then the random force will scale cubically with the initial deviations of the subnetwork from its steady state. One therefore expects an approximation error that is only of order δ^3, and the results shown in Fig. 4.12 are consistent with this. Importantly, the figure shows also that the approximation error is very much smaller in absolute terms, by four orders of magnitude at the largest initial deviations from steady state and more for smaller δ. We regard this as conclusive evidence that a quantitative description of subnetwork dynamics must include memory terms. The significant reduction in error over the linear projected equations, cf. Fig. 4.11, also emphasises that for quantitative accuracy nonlinearities in the memory have to be accounted for. By contrast, the memoryless approximations are hardly improved by the inclusion of nonlinear terms, which tells us that memory effects are crucial to get right first.

4.5 Summary

We have applied the projection method to a subnetwork embedded in the Shc-centred bulk of an EGFR signalling model. We illustrated how subnetwork-bulk interactions are illustrated in memory function amplitudes. We also analysed the timescales of the memory functions and how the memory function can be split into channels with different "source" and "receiver" channels dominating the memory effects. In Sec. 4.4 we showed the importance of the memory function in reducing the approximation error of the projected equations. The importance of memory functions in providing an accurate description of the dynamics of an oscillatory system can be seen in Appendix C.

Chapter 5

Michaelis-Menten dynamics in protein subnetworks

In Chapter 3, I derived the projected equations for protein reactions, which requires mass action kinetics. Reaction involving enzymes are a common type of biochemical reaction and are usually represented by Michaelis-Menten kinetics. If we are given a reaction system containing enzyme reactions and want to apply the projection method we therefore need to write the Michaelis-Menten dynamics as mass action dynamics. To do this we need to "create" enzymes and reaction rates that enable us to convert the Michaelis-Menten reactions into mass action form. This gives us a mass action system of equations to which we can apply the projection method, but is inconvenient both conceptually – we need to include extra species not present in the original system of reaction equations – and numerically, because the fast rates of enzyme reactions typically create a "stiff" system that has to be integrated using small timesteps. The aim of this chapter is, therefore, to find a way to derive the projected equations directly for the original reaction equations, without added enzyme species.

In Section 5.1.1, I give a summary of Michaelis-Menten dynamics and Section 5.1.2 explains in more detail under what conditions mass action enzyme kinetics can be accurately represented in Michaelis-Menten form: in essence these conditions are that the enzyme reactions must be fast, and the enzyme concentration low. Section 5.2 details our approach to obtaining projected equations for systems of reaction equations including

Michaelis-Menten reactions: we temporarily add enzymes to represent these reactions, derive the projected equations, and then take the limit of fast enzyme rate and low enzyme concentration. In this limit, I show that the added enzyme species can be eliminated in closed form. This leads to a simple procedure for constructing the projected equations without ever having to introduce additional species. I describe the approach separately for linearised (Sec. 5.3) and nonlinear (Sec. 5.4) dynamics, as the nonlinear case is more complicated technically but follows the same conceptual route as the linear one. Remarkably, even though Michaelis-Menten terms are generally nonlinear, I find that in the memory terms that are characteristic of the projected equations no additional nonlinearities appear, i.e. the memory terms involve linear concentration fluctuations for linearised dynamics, and linear and quadratic concentration fluctuations for nonlinear dynamics. Finally, in Section 5.5, I compare predictions from the original reaction equations with Michaelis-Menten dynamics to the projected equations with either enzymes explicitly retained or eliminated in closed form using the method derived in this paper. I find that the closed form elimination is both faster to evaluate computationally, and gives a more accurate approximation to the original reaction equations.

5.1 Michaelis-Menten dynamics

Many biochemical reactions are catalysed by enzymes. Generally each enzyme will enable a particular reaction without being consumed. A simple model of an enzyme reaction [54, 55] is written

$$u + e \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} c \xrightarrow{k_2} e + p \tag{5.1.1}$$

where u is a substrate (we use "u" not "s" here as we use the letter "s" to denote "subnetwork"), e is an enzyme, c is an enzyme-substrate complex and p is a product. The reaction rates are denoted k_1, k_{-1}, k_2. These reactions describe the binding of a free enzyme with a substrate to form a substrate-enzyme complex. This complex can then dissociate into enzyme and a product. In the traditional model substrate binding is reversible but product formation is not; we will consider also the more general case below, where both reactions are reversible.

5.1.1 Derivation of Michaelis-Menten equations

Let x_i be the concentration of species i. Then the set of mass actions equations for system (5.1.1) is

$$\begin{aligned}
\frac{\partial}{\partial t}x_u &= k_{-1}x_c - k_1 x_u x_e \\
\frac{\partial}{\partial t}x_e &= k_{-1}x_c - k_1 x_u x_e + k_2 x_c \\
\frac{\partial}{\partial t}x_c &= -k_{-1}x_c + k_1 x_u x_e - k_2 x_c \\
\frac{\partial}{\partial t}x_p &= k_2 x_c
\end{aligned} \tag{5.1.2}$$

From these equations it is easy to see that there is a conservation law between the enzyme and complex such that

$$\frac{\partial}{\partial t}x_e + \frac{\partial}{\partial t}x_c = 0 \Longrightarrow x_e + x_c = x_e^{\text{tot}} \tag{5.1.3}$$

The Michaelis-Menten description of the dynamics is obtained by exploiting the fact that enzyme reactions are typically fast. This allows one to reduce the system (5.1.1) to a simpler description where the enzyme and enzyme complex no longer appear explicitly.

Here we consider two different assumptions for achieving this simplification. The first is the "rapid equilibrium assumption" [55], which assumes that the rates k_1 and k_{-1} are both large while k_2 is smaller. Explicitly, we require $k_{-1} \gg k_2$ [56] so that the complex is much more likely to dissociate into enzyme and substrate than enzyme and product. The complex can then be taken to be in quasi-steady state with the substrate concentration at any given time, such that

$$k_1 x_u x_e = k_{-1} x_c \tag{5.1.4}$$

Substituting the enzyme conservation law (5.1.3) and solving for x_c we find

$$x_c = \frac{x_e^{\text{tot}} x_u}{K_d + x_u} \tag{5.1.5}$$

where $K_d = k_{-1}/k_1$ is known as the dissociation constant. The reaction flux for the system (5.1.1) is determined by the rate of product formation and is therefore $v = k_2 x_c$. Using equation (5.1.5) the reaction flux becomes

$$v = \frac{k_2 x_e^{\text{tot}} x_u}{K_d + x_u} = \frac{V_{\max} x_u}{K_d + x_u} \tag{5.1.6}$$

where

$$V_{\max} = k_2 x_e^{\text{tot}} \tag{5.1.7}$$

is the maximum reaction flux that can be achieved. The simplified description of the system (5.1.1) is then just $-(\partial/\partial t)x_u = (\partial/\partial t)x_p = v$.

The more common and less restrictive approach to simplifying the system of enzyme reactions is the general quasi-steady state assumption [57], where the dissociation of enzyme complex into enzyme and product is no longer taken as slow compared to dissociation into enzyme and substrate. Assuming all enzyme reaction rates are fast, the enzyme complex is then still in quasi-steady state at any time, but this steady state now depends on the concentrations of both substrate and product. Fig. 5.1 shows that the complex concentration rapidly reaches the quasi-steady state so that the assumption holds except for an initial transient when the enzyme concentration equilibrates. (For this and the other illustrative examples, we do not specify units explicitly but assume that times are measured in seconds and and concentrations in nMol per litre.) As before, we can then set the r.h.s. of the reaction equation for the enzyme complex to zero:

$$0 = -k_{-1}x_c + k_1 x_u x_e - k_2 x_c \tag{5.1.8}$$

Substituting in the enzyme conservation law (5.1.3) and solving we find that the steady state of the complex is

$$x_c = \frac{x_e^{\text{tot}} x_u}{(k_{-1} + k_2)/k_1 + x_u} \tag{5.1.9}$$

The reaction flux $v = k_2 x_c$ then becomes

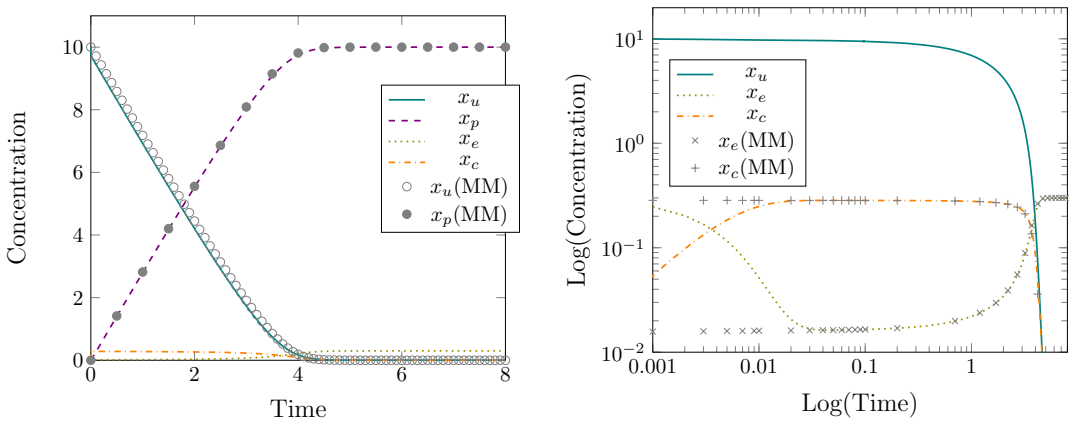
$$v = \frac{k_2 x_e^{\text{tot}} x_u}{(k_{-1} + k_2)/k_1 + x_u} = \frac{V_{\max} x_u}{K_{\text{m}} + x_u} \tag{5.1.10}$$

with

$$K_{\text{m}} = \frac{k_{-1} + k_2}{k_1} \tag{5.1.11}$$

the Michaelis constant. As before, the simplified description of the original reaction system can now be written in terms of the reaction flux v in (5.1.10), as

$$-\frac{\partial}{\partial t}x_u = \frac{V_{\max} x_u}{K_{\text{m}} + x_u} = \frac{\partial}{\partial t}x_p \tag{5.1.12}$$



(a) (b)

Figure 5.1: (a) Time courses of substrate, product, enzyme and complex found by solving the set of mass action equations (5.1.2) and the enzyme and complex time-courses found by applying the quasi-steady state assumption. We use the reaction rates $k_1 = 20, k_{-1} = 1, k_2 = 10$ as given in [58]; (b) shows the time courses with time and concentration on a log scale. The mass action time courses for the enzyme and complex are similar to the timecourses found from the quasi-steady state assumption except initially where the enzyme and complex have fast dynamics.

Fig. 5.2 illustrates the relationship between the reaction flux v and the substrate concentration.

We note that the assumption of a steady state with substrate and product is clearly more general than the rapid equilibrium assumption, and includes the latter as a limiting case: if $k_{-1} \gg k_2$, then $K_\mathrm{m} \approx K_\mathrm{d}$ and the results for the two approximations will be the same.

5.1.1.1 Reversible Michaelis-Menten dynamics

We discuss briefly how the above analysis is modified when there is a back reaction from the enzyme and product to the complex. In such cases one will need to use a somewhat more general reaction scheme

$$u + e \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} c \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} e + p \tag{5.1.13}$$

where the difference to the case discussed so far is the nonzero rate k_{-2}. The equation for the enzyme complex now has an extra contribution $k_{-2}x_e x_p$, while the equations for enzyme and product contain the same additional term with a negative sign. Using the

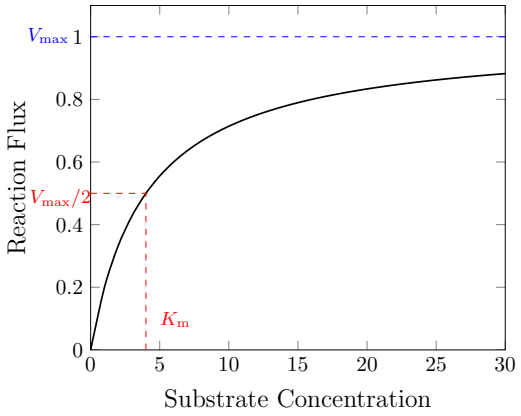


Figure 5.2: The relationship between the reaction flux and substrate concentration for a single Michaelis-Menten reaction. $V_{\max}$ is the maximum reaction flux obtainable from substrate to product. K_{m} is the concentration that produces half this flux. $K_{\mathrm{m}} = 4$ and $V_{\max} = 1$ as taken from [58].

quasi-steady state assumption one then obtains a reaction flux of the form

$$v = \frac{V_{\mathrm{u}} x_u / K_{\mathrm{u}} - V_{\mathrm{p}} x_p / K_{\mathrm{p}}}{1 + x_u / K_{\mathrm{u}} + x_p / K_{\mathrm{p}}} \tag{5.1.14}$$

where V_{u} and V_{p} are the maximum reaction rates for the forward and reverse reactions respectively and are given by

$$V_{\mathrm{u}} = k_2 x_e^{\mathrm{tot}}, \quad V_{\mathrm{p}} = k_{-1} x_e^{\mathrm{tot}} \tag{5.1.15}$$

Similarly K_{u} and K_{p} are the Michaelis constants for the forward and reverse reactions and are written in terms of the mass action rates as

$$K_{\mathrm{u}} = \frac{k_2 + k_{-1}}{k_1}, \quad K_{\mathrm{p}} = \frac{k_2 + k_{-1}}{k_{-2}} \tag{5.1.16}$$

5.1.2 Quantitative accuracy of Michaelis-Menten approximation

We have shown above the assumptions used to obtain Michaelis-Menten dynamics (5.1.12) – or its generalisation (5.1.14) – and now want to look at the quantitative accuracy of these descriptions. In particular, we want to understand under what conditions on the mass action parameters the Michaelis-Menten approximation becomes exact, so that we can later take an appropriate limit for these parameters in the construction of the projected equations.

We follow the work of Segel and Slemrod [59] who found a small parameter that is needed to allow the quasi-steady state assumption to hold. They calculate this parameter by using a singular perturbation approach on a dimensionless form of the reaction equations.

We focus first on the traditional case of no back reaction from enzyme and product to complex. Here we have four mass action parameters k_1, k_{-1}, k_2 and x_e^{tot}. These are linked by the two parameters appearing in the Michaelis-Menten description: the maximum reaction flux $V_{\max}$ (5.1.7) and the Michaelis constant K_{m} (5.1.11). Hence there are two free parameters that we want to choose to make the Michaelis-Menten representation of the enzyme reaction exact.

One condition for exactness is that the enzyme reactions are fast enough to justify the quasi-steady state assumption for the enzyme and enzyme complex. More precisely, the rate of change of the concentration of the enzyme complex needs to be much faster than the rate of change of the substrate and product concentrations. Substituting the conservation law (5.1.3) into the mass action equation for the complex we find that

$$\begin{aligned}\frac{\partial}{\partial t}x_c &= -k_{-1}x_c + k_1x_u(x_e^{\text{tot}} - x_c) + k_2x_c \\ &= -(x_c - x_c^*)(k_1x_u + k_{-1} + k_2)\end{aligned} \tag{5.1.17}$$

where x_c^* is the quasi-steady state value of x_c as given in (5.1.9). On the other hand the substrate equation evolves, if the Michaelis-Menten description is accurate, according to (5.1.12). Written out explicitly this reads

$$\frac{\partial}{\partial t}x_u = -\frac{k_1x_ux_e^{\text{tot}}k_2}{k_1x_u + k_{-1} + k_2} \tag{5.1.18}$$

The ratio for the rate of change of substrate over the rate of change of complex is therefore

$$\begin{aligned}\frac{\frac{\partial}{\partial t}x_u/x_u}{\frac{\partial}{\partial t}x_c/(x_c - x_c^*)} &= \frac{k_1x_e^{\text{tot}}k_2}{(k_1x_u + k_{-1} + k_2)^2} \\ &= \frac{V_{\max}}{k_1(x_u + K_{\text{m}})^2} \\ &= \frac{x_e^{\text{tot}}}{(x_u + K_{\text{m}})}\frac{k_2}{k_1(x_u + K_{\text{m}})}\end{aligned} \tag{5.1.19}$$

This ratio needs to be small to ensure that the quasi-steady state approximation is reasonable; the approximation will be exact in the limit when the ratio tends to zero. We can see that $x_e^{\text{tot}}/(x_u + K_{\text{m}})$ is the small parameter from Segel and Slemrod [59].

We note further that

$$\frac{k_2}{k_1(x_u + K_\mathrm{m})} = \frac{k_2}{k_2 + k_{-1}} \cdot \frac{K_\mathrm{m}}{x_u + K_\mathrm{m}} \tag{5.1.20}$$

where both terms on the right hand side are less than 1 and therefore the ratio for the change of substrate over the rate of change of complex is small provided that $x_e^{\mathrm{tot}}/(x_u + K_\mathrm{m})$ is small.

We also note that in writing down (5.1.12) one implicitly assumes a conservation law of the form $x_u + x_p = \text{const}$. However, the corresponding true conservation law of the mass action kinetics (5.1.2) is $x_u + x_c + x_p = \text{const}$. The Michaelis-Menten description therefore requires a small concentration of enzyme complex. From (5.1.9) this is of the order of x_e^{tot}, more specifically we require that $x_e^{\mathrm{tot}} x_u/(K_\mathrm{m} + x_u) \ll x_u$. This implies that $x_e^{\mathrm{tot}}/(K_\mathrm{m} + x_u) \ll 1$ which is again the condition given by Segel and Slemrod [59] to ensure that the quasi-steady state assumption is valid. From the constancy of $V_{\max} = k_2 x_e^{\mathrm{tot}}$ in (5.1.7), it then follows that k_2 must be large in inverse proportion to x_e^{tot} being small.

The above considerations suggest that we should rewrite the enzyme reaction rates and total enzyme concentrations as

$$k_1 = \bar{k}_1\gamma,\ k_{-1} = \bar{k}_{-1}\gamma,\ k_2 = \bar{k}_2\gamma,\ x_e^{\mathrm{tot}} = \bar{x}_e^{\mathrm{tot}}/\gamma \tag{5.1.21}$$

where γ is a fast rate parameter. To fix the scale of this parameter, we choose $\bar{k}_2 = 1$ in the following, so that γ is just the reaction rate k_2. The definitions of $V_{\max}$ (5.1.7) and K_m (5.1.11) then allow us to write the remaining parameters as

$$\begin{aligned} \bar{x}_e^{\mathrm{tot}} &= V_{\max} \\ \bar{k}_1 &= \frac{1 + \bar{k}_{-1}}{K_\mathrm{m}} \end{aligned}$$

As expected we now have *two* variables, $\bar{k}_{-1}$ and γ, that parameterise the possible mass action kinetics underlying a given Michaelis-Menten reaction. Our reasoning above suggests that the Michaelis-Menten description will become exact for $\gamma \to \infty$, irrespective of the value of $\bar{k}_{-1}$ [60].

To test this statement numerically, we vary $\bar{k}_{-1}$ and γ to see when the substrate and product concentrations $\hat{x}(t)$ predicted by the Michaelis-Menten equation give good approximations to the values $x(t)$ from the underlying mass action kinetics. Let us define

the accuracy of the approximation as

$$\Delta = \frac{1}{T}\int_0^T dt' \, \frac{1}{N}\sum_{i=1}^{N} |x_i(t) - \hat{x}_i(t)| \tag{5.1.22}$$

where here the sum runs over $i = u, p$ ($N = 2$) and the total time interval for the error measurement is chosen such as to capture the transient dynamics in the approach to the overall system steady state. Then Fig. 5.3 shows that for the simple model of enzyme interaction defined in (5.1.1) with rate parameters $K_{\mathrm{m}} = 4$ and $V_{\max} = 1$, the Michaelis-Menten description does indeed become increasingly accurate when we increase γ, with Δ decreasing as $\sim \gamma^{-1}$. Changing $\bar{k}_{-1}$, on the other hand, has only a very weak effect on the quality of the approximation. This was expected, as the condition given by Segel and Slemrod [59] for the quasi-steady state condition to hold is both necessary and sufficient and is not dependent on the size of k_{-1}. Note also that $\bar{k}_{-1} \gg 1$ means that $k_{-1} \gg k_2$, which is the condition required for rapid equilibrium (between enzyme complex and substrate) as discussed in Section 5.1. For finite $\bar{k}_{-1}$, on the other hand, the dissociation rates of the enzyme complex into enzyme and substrate or enzyme and complex are comparable so one has the more general quasi-steady state (between enzyme complex, and substrate and product) scenario.

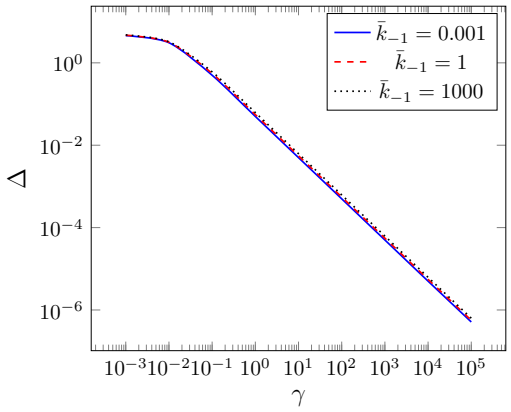


Figure 5.3: Plot of total error obtained from comparing the timecourses of the original Michaelis-Menten reaction equations from example (5.1.1) with the mass action description obtained from varying the value of the fast rate γ. Reaction rates are $K_{\mathrm{m}} = 4$ and $V_{\max} = 1$ and initial condition $x_u(0) = 10$, with error measured over the time interval $T = 100$.

We comment briefly on how realistic the requirement of large γ is for enzyme reactions. The reaction rate k_2 is also known as k_{cat} and from (5.1.7) represents the turnover

number of an enzyme. This is the number of substrate molecules converted into product by an enzyme molecule in a unit time when the enzyme is fully saturated with substrate. The turnover numbers of most enzymes range from 1 to 10^4 per second [61] and certainly at the upper end of this range are significantly larger than typical reaction rates for protein reactions, which are of the order of 1 per second or less. For example, for the reactions in Kholodenko et al. [13] the reactions range from 0.6 to $1.5 \cdot 10^{-3}\mathrm{s}^{-1}$. In Fig. 5.3, the slow reaction rate is the one for the substrate. Defining this as before as $(\partial x_u/\partial t)/x_u$, we find the slow "protein" rate is of the order of – in fact bounded by – $V_{\max}/K_{\mathrm{m}}$. In the figure we see small approximation errors from Michaelis-Menten for γ above ca. 10^3, which is $\sim 10^4$ times larger than the slow rate.

The above considerations apply where there is no back reaction from product and enzyme to form an enzyme complex. In the more general case where such a reaction is possible with rate constant k_{-2}, one can again argue that the enzyme reaction rates must be large and the enzyme concentration small, and so write these parameters in terms of a fast rate γ exactly as in (5.1.21). The difference is that because we now have one more mass action rate, but *two* more Michaelis-Menten parameters for the back reaction, there is only one free parameter in the mass action parameters, namely γ. If we again let $\bar{k}_2 = 1$ we find explicitly for the remaining scaled mass action parameters

$$\begin{aligned}
\bar{x}_e^{\mathrm{tot}} &= V_{\mathrm{u}} \\
\bar{k}_{-1} &= \frac{V_{\mathrm{p}}}{V_{\mathrm{u}}} \\
\bar{k}_1 &= \frac{1 + V_{\mathrm{p}}/V_{\mathrm{u}}}{K_{\mathrm{u}}} \\
\bar{k}_{-2} &= \frac{1 + V_{\mathrm{p}}/V_{\mathrm{u}}}{K_{\mathrm{p}}}
\end{aligned}$$

One then expects that the Michaelis-Menten description will be an accurate approximation of the mass action kinetics when γ is large enough, and this is indeed what we find in numerical tests (not shown).

5.2 Enzyme reactions in the projected equations

The projection method as applied to protein interaction networks [62] works with mass action kinetics. Therefore if we are given an interaction network that includes enzyme

reactions in Michaelis-Menten form, then *a priori* we need to represent these reactions in explicit mass action form to allow us to compute the projected equations. As explained in the introduction, this is a disadvantage both conceptually and computationally. Our aim here is to implement this approach *analytically* instead: we add enzyme species to the network to get mass action kinetics and take the large enzyme rate and low enzyme concentration limit in which the mass action dynamics becomes exactly identical to Michaelis-Menten dynamics. The challenge is to understand what happens in this limit to the projected equations, which will then give us a method for constructing the limiting equations directly, without ever representing enzymes explicitly.

The aim of the projection method generally is to provide a description of the dynamics of a protein interaction subnetwork embedded in a bulk network. The Zwanzig-Mori projection method [27] can be used to obtain such a description, specifically equations for the time evolution of the protein concentrations in a chosen subnetwork. Full details of the projection method applied to protein interaction networks are given in Chapters 2 and 3. We summarise below the features necessary for the analysis of enzyme dynamics.

The time evolution of any observable $a(\boldsymbol{\delta x}, t)$ is given by $(\partial/\partial t)a = \mathcal{L}a$, where $\mathcal{L}$ is the adjoint Fokker-Planck operator, whose drift term contains information from the mass action equations. We showed in Chapter 3 that if we focus on a set of observables $\boldsymbol{z}$ containing $\boldsymbol{\delta x}$ and all products like δx_1^2 and $\delta x_1 \delta x_2$, then the operator $\mathcal{L}$ can be written in *matrix form* $\boldsymbol{L}$ such that

$$\frac{\partial}{\partial t} z_\alpha = \sum_\beta z_\beta L_{\beta\alpha} + \delta x^3 + \mathcal{O}(\epsilon) \tag{5.2.1}$$

To analyse subnetwork dynamics one splits this matrix into subnetwork and bulk sections so that

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}^{\mathrm{s,s}} & \boldsymbol{L}^{\mathrm{s,b}} & 0 & 0 & 0 \\ \boldsymbol{L}^{\mathrm{b,s}} & \boldsymbol{L}^{\mathrm{b,b}} & 0 & 0 & 0 \\ \boldsymbol{L}^{\mathrm{ss,s}} & \boldsymbol{L}^{\mathrm{ss,b}} & \boldsymbol{L}^{\mathrm{ss,ss}} & \boldsymbol{L}^{\mathrm{ss,sb}} & \boldsymbol{L}^{\mathrm{ss,bb}} \\ \boldsymbol{L}^{\mathrm{sb,s}} & \boldsymbol{L}^{\mathrm{sb,b}} & \boldsymbol{L}^{\mathrm{sb,ss}} & \boldsymbol{L}^{\mathrm{sb,sb}} & \boldsymbol{L}^{\mathrm{sb,bb}} \\ \boldsymbol{L}^{\mathrm{bb,s}} & \boldsymbol{L}^{\mathrm{bb,b}} & \boldsymbol{L}^{\mathrm{bb,ss}} & \boldsymbol{L}^{\mathrm{bb,sb}} & \boldsymbol{L}^{\mathrm{bb,bb}} \end{pmatrix} \tag{5.2.2}$$

where, for example, $\boldsymbol{L}^{\mathrm{b,s}}$ contains the linear coefficients of bulk concentrations in the equations of motion for subnetwork concentrations, while the coefficients of subnetwork-bulk products in these equations are in the block $\boldsymbol{L}^{\mathrm{sb,s}}$.

If we now let $\{a_\alpha(\boldsymbol{\delta x})\}$ be a set of observables from the subnetwork, the projected equations have the form

$$\frac{\partial}{\partial t}a_\alpha(t) = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \int_0^t dt' \sum_\beta a_\beta(t')M_{\beta\alpha}(t-t') + r_\alpha(t) \tag{5.2.3}$$

We choose specifically as subnetwork observables all the subnetwork concentrations and their products, i.e. the entries of $\boldsymbol{z}$ only involving the subnetwork. We denote these collectively by S, and the remaining species – all bulk concentrations (b) and concentration products involving the bulk (sb and bb) – by the letter B. Then the "rate matrix" $\boldsymbol{\Omega}$ in (5.2.3) is simply [62]

$$\boldsymbol{\Omega} = \boldsymbol{L}^{\mathrm{S,S}} = \begin{pmatrix} \boldsymbol{L}^{\mathrm{s,s}} & 0 \\ \boldsymbol{L}^{\mathrm{ss,s}} & \boldsymbol{L}^{\mathrm{ss,ss}} \end{pmatrix} \tag{5.2.4}$$

It contains terms from the subnetwork dynamics that are local in time. The memory function (matrix) can be written as [62]

$$\boldsymbol{M}(\Delta t) = \boldsymbol{L}^{\mathrm{S,B}} e^{\boldsymbol{L}^{\mathrm{B,B}}\Delta t}\boldsymbol{L}^{\mathrm{B,S}} \tag{5.2.5}$$

where $\Delta t = t - t'$. The entries of this matrix are the memory functions $M_{\beta\alpha}(\Delta t)$ and determine how strongly the past values of the observable a_β affect the present rate of change of a_α. For later use we note here the Laplace transform version of the memory function, which is

$$\hat{\boldsymbol{M}}(z) = \boldsymbol{L}^{\mathrm{S,B}}\left(z - \boldsymbol{L}^{\mathrm{B,B}}\right)^{-1}\boldsymbol{L}^{\mathrm{B,S}} \tag{5.2.6}$$

One important property of the memory functions is their boundary structure: if we define a boundary species as a subnetwork species that directly interacts with the bulk, then among the projected equations for subnetwork concentrations only those of boundary species contain memory terms.

The final term in (5.2.3), $r(t)$, is what is known as the random force. It accounts for the fact that because of the interaction between subnetwork and bulk, the time evolution of the subnetwork observables cannot be closed.

Our analysis starts from a given reaction network involving unary and binary protein reactions, and enzyme reactions described by Michaelis-Menten terms. We assume for simplicity that all enzyme reactions are reversible; the irreversible scenario can be obtained from this as the limiting case where the rate of dissociation into enzyme and

product is much larger than the rate for formation of enzyme complex in the reverse direction. Using the conservation law $\delta x_e + \delta x_c = 0$ we can eliminate the enzyme complex from the description of each enzyme reaction, and write the mass-action kinetics as

$$\begin{aligned} \frac{\partial}{\partial t}\delta x_u &= -f_{ue,c} + \dots \\ \frac{\partial}{\partial t}\delta x_e &= -f_{ue,c} - f_{pe,c} \\ \frac{\partial}{\partial t}\delta x_p &= -f_{pe,c} + \dots \end{aligned} \tag{5.2.7}$$

where the dots indicate fluxes from other reactions. The reaction fluxes from substrate and enzyme to complex, and from product and enzyme to complex, read respectively

$$\begin{aligned} f_{ue,c} &= k^-_{c,ue}(-x_e^{\rm tot} + y_e + \delta x_e) + k^+_{ue,c}(y_u + \delta x_u)(y_e + \delta x_e) \\ f_{pe,c} &= k^-_{c,pe}(-x_e^{\rm tot} + y_e + \delta x_e) + k^+_{pe,c}(y_p + \delta x_p)(y_e + \delta x_e) \end{aligned} \tag{5.2.8}$$

where $k^-_{c,ue}$ is the rate of dissociation of the complex into substrate and enzyme while $k^+_{ue,c}$ is the rate of complex formation from substrate and enzyme, with analogous definitions for the rates involving the product. In writing the fluxes we have used the enzyme conservation law (5.1.3) to eliminate the complex concentration via $x_c = x_e^{\rm tot} - x_e = x_e^{\rm tot} - y_e - \delta x_e$. The enzyme steady state concentration y_e can be found by requiring that in the steady state $\delta x_e = \delta x_u = \delta x_p = 0$, the two fluxes must sum to zero to ensure $(\partial/\partial t)\delta x_e = 0$. This gives

$$y_e = x_e^{\rm tot}\,\frac{k^-_{c,ue} + k^-_{c,pe}}{k^-_{c,ue} + k^-_{c,pe} + k^+_{ue,c}y_u + k^+_{pe,c}y_p} \tag{5.2.9}$$

We now write the enzymatic reaction rates in terms of a fast rate γ as in (5.1.21), and similarly the steady state enzyme concentration, which from (5.2.9) must scale as the inverse of γ, i.e. $y_e = \bar{y}_e/\gamma$, like the total enzyme concentration $x_e^{\rm tot}$. This gives

$$\begin{aligned} f_{ue,c} &= \bar{k}^-_{c,ue}\gamma(-\bar{x}_e^{\rm tot}/\gamma + \bar{y}_e/\gamma + \delta x_e) + \bar{k}^+_{ue,c}\gamma(y_u + \delta x_u)(\bar{y}_e/\gamma + \delta x_e) \\ f_{pe,c} &= \bar{k}^-_{c,pe}\gamma(-\bar{x}_e^{\rm tot}/\gamma + \bar{y}_e/\gamma + \delta x_e) + \bar{k}^+_{pe,c}\gamma(y_p + \delta x_p)(\bar{y}_e/\gamma + \delta x_e) \end{aligned} \tag{5.2.10}$$

The scaled rates and steady state enzyme concentration are related to the Michaelis-Menten parameters as explained after (5.1.13), i.e.

$$\begin{aligned} V_{\rm u} &= \frac{\bar{k}^-_{c,pe}\bar{y}_e(\bar{k}^-_{c,ue} + \bar{k}^-_{c,pe} + \bar{k}^+_{ue,c}y_u + \bar{k}^+_{pe,c}y_p)}{\bar{k}^-_{c,ue} + \bar{k}^-_{c,pe}}, \quad K_{\rm u} = \frac{\bar{k}^-_{c,pe} + \bar{k}^-_{c,ue}}{\bar{k}^+_{ue,c}} \\ V_{\rm p} &= \frac{\bar{k}^-_{c,ue}\bar{y}_e(\bar{k}^-_{c,ue} + \bar{k}^-_{c,pe} + \bar{k}^+_{ue,c}y_u + \bar{k}^+_{pe,c}y_p)}{\bar{k}^-_{c,ue} + \bar{k}^-_{c,pe}}, \quad K_{\rm p} = \frac{\bar{k}^-_{c,pe} + \bar{k}^-_{c,ue}}{\bar{k}^+_{pe,c}} \end{aligned} \tag{5.2.11}$$

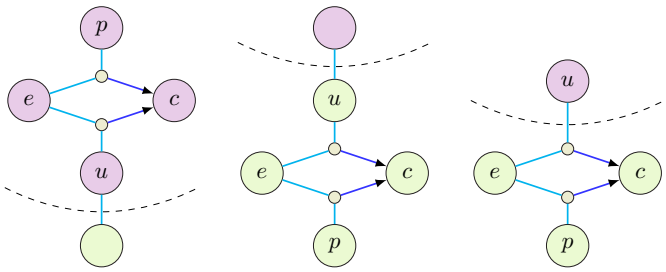
In the above representation it is not obvious which terms have to be regarded as "fast" in the remainder of the analysis, and which as slow. We therefore switch to dimensionless concentration variables $\delta\tilde{x}_i = \delta x_i / y_i$. In terms of these we have

$$\begin{aligned}
\frac{\partial}{\partial t}\delta\tilde{x}_u &= -f_{ue,c}/y_u + \dots \\
\frac{\partial}{\partial t}\delta\tilde{x}_e &= -\gamma(f_{ue,c} + f_{pe,c})/\bar{y}_e \\
\frac{\partial}{\partial t}\delta\tilde{x}_p &= -f_{pe,c}/y_p + \dots
\end{aligned} \tag{5.2.12}$$

with enzymatic fluxes

$$\begin{aligned}
f_{ue,c} &= \bar{k}^-_{c,ue}\bar{y}_e(-\bar{x}^{\mathrm{tot}}_e/\bar{y}_e + 1 + \delta\tilde{x}_e) + \bar{k}^+_{ue,c}y_u\bar{y}_e(1+\delta\tilde{x}_u)(1+\delta\tilde{x}_e) \\
f_{pe,c} &= \bar{k}^-_{c,pe}\bar{y}_e(-\bar{x}^{\mathrm{tot}}_e/\bar{y}_e + 1 + \delta\tilde{x}_e) + \bar{k}^+_{pe,c}y_p\bar{y}_e(1+\delta\tilde{x}_p)(1+\delta\tilde{x}_e)
\end{aligned} \tag{5.2.13}$$

Here one sees clearly that the enzyme evolution equation contains only fast terms that scale with γ, while the equations for substrate and product only contain slow terms. We will therefore use dimensionless concentrations throughout, and to lighten the notation we will in the following drop the tildes, as well as the bars indicating rescaling with γ. Note also that as at steady state ($\boldsymbol{\delta x} = 0$) the total flux into or out of any molecular species must be zero, we can drop the constant pieces from $f_{ue,c}$ and $f_{pe,c}$: in the equation for δx_e they have to cancel against each other, and against the other steady state fluxes in the equations for δx_u and δx_p.



(a) subnetwork (b) bulk (c) boundary

Figure 5.4: The three possible cases we are considering are (a) enzyme reactions fully contained in the subnetwork; (b) enzyme reactions fully contained in the bulk; (c) enzyme reactions on the boundary, with the substrate in the subnetwork and the enzyme, complex and product in the bulk.

Our task now is to take a mass-action reaction system where every enzyme reaction is represented as in (5.2.13), and to find closed form expressions for the rate matrix and

memory functions in the limit $\gamma \to \infty$ where we know that this mass-action description becomes identical to Michaelis-Menten dynamics. The effect of the enzyme reactions depends on where they are located relative to subnetwork and bulk, with three potentially distinct cases as shown in Fig. 5.4. Enzymes in the subnetwork (Fig. 5.4a) are by definition away from the boundary of the subnetwork. We will therefore find "fast" equations of motion for them without any memory terms. Such enzymes can then be kept explicitly in a first stage of our analysis, and eliminated in a second stage following the standard logic that leads to the Michaelis-Menten description.

Bulk enzyme reactions (Fig. 5.4b) sit entirely within the bulk, contributing fast reaction rates to $\boldsymbol{L}^{\mathrm{B,B}}$. Accordingly we will find that they only give contributions to the memory functions, not the rate matrix.

The third case is the one where the substrate is on the boundary and the enzyme is in the bulk. Such enzyme reactions, whose rates appear in $\boldsymbol{L}^{\mathrm{S,B}}$, $\boldsymbol{L}^{\mathrm{B,B}}$ and $\boldsymbol{L}^{\mathrm{B,S}}$, will contribute fast terms in the memory functions that decay on a timescale of order $1/\gamma$. In the limit $\gamma \to \infty$ these terms become local in time and so turn into contributions to the rate matrix.

5.3 Linearised Dynamics

In linearised dynamics we only consider terms in the mass-action kinetics up to linear order in $\boldsymbol{\delta x}$. The dimensionless scaled reaction equations for a reversible Michaelis-Menten reaction are then, from (5.2.12) and (5.2.13),

$$\begin{aligned}
\frac{\partial}{\partial t}\delta x_u &= -k^-_{c,ue}(y_e/y_u)\delta x_e - k^+_{ue,c}y_e(\delta x_u + \delta x_e) + \dots \\
\frac{\partial}{\partial t}\delta x_e &= -\gamma[(k^-_{c,ue} + k^-_{c,ep})\delta x_e + k^+_{ue,c}y_u(\delta x_u + \delta x_e) + k^+_{ep,c}y_p(\delta x_p + \delta x_e)] \\
\frac{\partial}{\partial t}\delta x_p &= -k^-_{c,ep}(y_e/y_p)\delta x_e - k^+_{ep,c}y_e(\delta x_p + \delta x_e) + \dots
\end{aligned} \tag{5.3.1}$$

Let us partition the matrix form of the adjoint Fokker-Planck matrix operator (5.2.2) so that the bulk species are split into fast and slow blocks. If e′ and e represent the collection of subnetwork and bulk enzymes respectively and s and b represent the other

molecular species in the subnetwork and bulk, we partition as

$$\boldsymbol{L} = \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{S,S}} & \boldsymbol{L}^{\mathrm{S,B}} \\ \hline \boldsymbol{L}^{\mathrm{B,S}} & \boldsymbol{L}^{\mathrm{B,B}} \end{array}\right) = \left(\begin{array}{cc:cc} \boldsymbol{L}^{\mathrm{s,s}} & \boldsymbol{L}^{\mathrm{s,e'}} & \boldsymbol{L}^{\mathrm{s,b}} & \boldsymbol{L}^{\mathrm{s,e}} \\ \boldsymbol{L}^{\mathrm{e',s}} & \boldsymbol{L}^{\mathrm{e',e'}} & \boldsymbol{L}^{\mathrm{e',b}} & \boldsymbol{L}^{\mathrm{e',e}} \\ \hdashline \boldsymbol{L}^{\mathrm{b,s}} & \boldsymbol{L}^{\mathrm{b,e'}} & \boldsymbol{L}^{\mathrm{b,b}} & \boldsymbol{L}^{\mathrm{b,e}} \\ \boldsymbol{L}^{\mathrm{e,s}} & \boldsymbol{L}^{\mathrm{e,e'}} & \boldsymbol{L}^{\mathrm{e,b}} & \boldsymbol{L}^{\mathrm{e,e}} \end{array}\right) = \left(\begin{array}{c:cc} m & w_1 & f_1 \\ \hdashline w_2 & w_3 & f_2 \\ w_4 & w_5 & f_3 \end{array}\right) \tag{5.3.2}$$

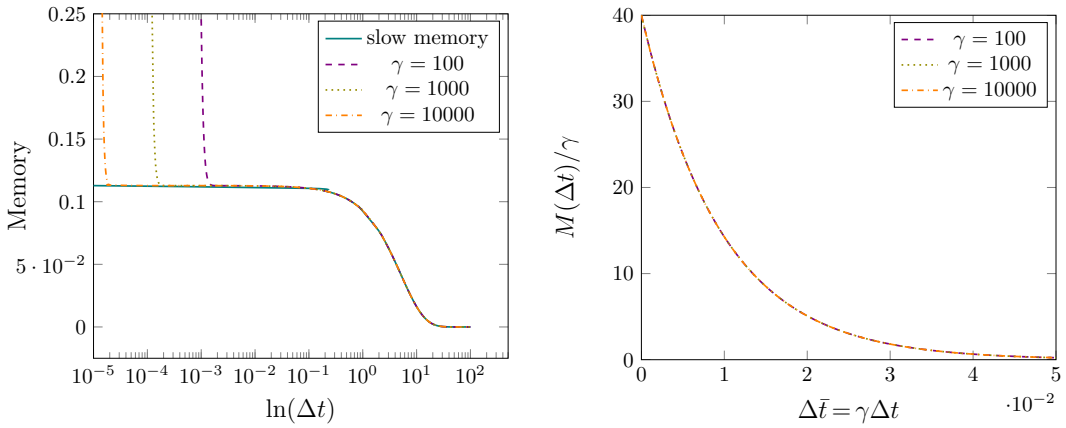
where w are slow terms and f are fast terms; the top left block denoted m contains a mixture of fast and slow terms. In writing the last equality above we have grouped s and e$'$ together; the resulting specific 3×3 block structure of slow and fast terms is one that we will find again in the case of the full nonlinear dymamics. Note that because subnetwork enzymes only have interactions with subnetwork species (s and e$'$), $\boldsymbol{L}^{\mathrm{b,e'}}$ and $\boldsymbol{L}^{\mathrm{e,e'}}$ are zero. Similarly, because bulk proteins or enzymes do not interact with subnetwork enzymes, $\boldsymbol{L}^{\mathrm{e',b}}$ and $\boldsymbol{L}^{\mathrm{e',e}}$ vanish. This means that subnetwork enzymes do not feature at all in the calculation of the memory function (5.2.6), which makes intuitive sense. The vanishing of $\boldsymbol{L}^{\mathrm{b,e'}}$ and $\boldsymbol{L}^{\mathrm{e,e'}}$ is important also as these blocks contain rates for the time evolution of (subnetwork) enzymes, which by our construction scale with γ: if these blocks were nonzero, it would change the character of w_2 and w_4 from slow to fast.

To analyse the memory function (5.2.5) that results from (5.3.2), we note that $\boldsymbol{L}^{\mathrm{B,B}}$ has both slow and fast sub-blocks. As a result the memory function should have both slow contributions that decay on $O(1)$ timescales, and fast contributions that decay for time differences of $O(1/\gamma)$. As the memory function appears as a weight in an integral over the past (5.2.3), the fast contributions only matter for $\gamma \to \infty$ if their amplitude is proportional to γ so that the integral over all time differences remains finite. Accounting also for subleading terms in the amplitude dependence then suggests the following decomposition of the memory function:

$$M(\Delta t) = (\gamma M_f^0(\Delta\bar{t}) + M_f^1(\Delta\bar{t}) + \ldots) + (M_w^0(\Delta t) + \frac{1}{\gamma} M_w^1(\Delta t) + \ldots) \tag{5.3.3}$$

where $\Delta\bar{t} = \gamma\Delta t$. In principle an arbitrary constant can be added to e.g. $M_f^1(\Delta\bar{t})$ and subtracted from $M_w^0(\Delta t)$; we fix this constant by requiring that all fast contributions decay to zero for large $\Delta\bar{t}$. Fig. 5.5 shows an example of the above decomposition.

The leading fast and slow contributions can now be extracted relatively simply from the



(a) (b)

Figure 5.5: Example plots of a memory function when the enzyme is on the boundary for different values of γ. (a) The memory has a slow and a fast part, with the fast part moving to shorter and shorter timescales as γ increases. (b) Scaled plot of the fast part: if the fast memory is divided by γ we see that a scaling plot is approached for large γ. This shows that the amplitude of the fast part of the memory grows proportionally to γ.

Laplace transform of the memory function (5.2.6), which has the decomposition

$$\hat{M}(z) = (\hat{M}_f^0(\bar{z}) + \frac{1}{\gamma}\hat{M}_f^1(\bar{z}) + \ldots) + (\hat{M}_w^0(z) + \frac{1}{\gamma}\hat{M}_w^1(z) + \ldots) \tag{5.3.4}$$

where $\bar{z} = z/\gamma$. The leading fast term can be extracted by taking the limit

$$\lim_{\gamma\to\infty} \hat{M}(z)\big|_{\bar{z}=\text{const}} = \hat{M}_f^0(\bar{z}) \tag{5.3.5}$$

because the subleading fast terms are down by powers of $1/\gamma$, and in the slow terms $z = \gamma\bar{z} \to \infty$ so that the Laplace transforms $\hat{M}_w^0(z)$ etc. vanish. The leading slow part can then be found from a limit at constant z, namely

$$\lim_{\gamma\to\infty} \left[\hat{M}(z) - \hat{M}_f^0(0)\right]\Big|_{z=\text{const}} = \hat{M}_w^0(z) \tag{5.3.6}$$

We could have equivalently written $\hat{M}_f^0(\bar{z})$ inside the square brackets, as $\bar{z} = z/\gamma \to 0$ when $\gamma \to \infty$ at fixed z.

Using the method above, we can now find the fast and slow pieces of the memory function derived from the adjoint Fokker-Planck (matrix) operator in equation (5.3.2).

From (5.2.6) this memory function reads

$$\hat{M}(z) = \begin{pmatrix} w_1 & f_1 \end{pmatrix} \begin{pmatrix} z - w_3 & -f_2 \\ -w_5 & z - f_3 \end{pmatrix}^{-1} \begin{pmatrix} w_2 \\ w_4 \end{pmatrix} \tag{5.3.7}$$

and after using the Woodbury identity [63] and simplifying we obtain

$$\begin{aligned} \hat{M}(z) = &\left(w_1 - f_1(z - f_3)^{-1}(-w_5)\right)\left(z - w_3 + f_2(z - f_3)^{-1}w_5\right)^{-1}\left(w_2 + f_2(z - f_3)^{-1}w_4\right) \\ &+ f_1(z - f_3)^{-1}w_4 \end{aligned} \tag{5.3.8}$$

If we now write all fast blocks as $f = \gamma\bar{f}$, we can see that application of (5.3.5) identifies the fast part of the memory function as

$$\hat{M}_f^0(\bar{z}) = \bar{f}_1(\bar{z} - \bar{f}_3)^{-1}w_4 \tag{5.3.9}$$

while (5.3.6) gives for the slow part

$$\hat{M}_w^0(z) = (w_1 - \bar{f}_1(\bar{f}_3)^{-1}w_5)(z - w_3 - \bar{f}_2(\bar{f}_3)^{-1}w_5)^{-1}(w_2 - \bar{f}_2(\bar{f}_3)^{-1}w_4) \tag{5.3.10}$$

Here we have used the fact that in the combination $z - \gamma\bar{f}_3$, the first term can be neglected when $\gamma \to \infty$ at constant z.

The fast part of the memory decays on an ever shorter timescale $\sim 1/\gamma$ as γ increases. In the limit, when it is used inside a memory function integral, it becomes equivalent to a delta function $\delta(\Delta t)$ multiplied by the area under the fast piece, which is just $\hat{M}_f^0(0) = -\bar{f}_1(\bar{f}_3)^{-1}w_4$. The rate matrix that we obtain from (5.3.2) for $\gamma \to \infty$ is therefore

$$\boldsymbol{\Omega} = m - \bar{f}_1(\bar{f}_3)^{-1}w_4 \tag{5.3.11}$$

while the memory function in the same limit is given by (5.3.10). We can now compare to the rate matrix and memory function that would result from an adjoint Fokker-Planck operator like (5.3.2) without the fast bulk variables e, i.e.

$$\boldsymbol{L}_{\backslash \mathrm{e}} = \left(\begin{array}{c|c} m & w_1 \\ \hline w_2 & w_3 \end{array}\right) \tag{5.3.12}$$

This would give $\Omega = m$ and $\hat{M}(z) = w_1(z - w_3)^{-1}w_2$. Looking at (5.3.10) and (5.3.11), we conclude that the large-γ limit gives results that can equivalently be obtained by

using an adjoint Fokker-Planck matrix without the fast bulk variables that is modified from $\boldsymbol{L}_{\backslash \mathrm{e}}$ to $\boldsymbol{L}_{\mathrm{eff}} = \boldsymbol{L}_{\backslash \mathrm{e}} + \Delta\boldsymbol{L}_{\backslash \mathrm{e}}$, where

$$\Delta\boldsymbol{L}_{\backslash \mathrm{e}} = -\left(\begin{array}{c|cc} \bar{f}_1(\bar{f}_3)^{-1}w_4 & \bar{f}_1(\bar{f}_3)^{-1}w_5 \\ \hline \bar{f}_2(\bar{f}_3)^{-1}w_4 & \bar{f}_2(\bar{f}_3)^{-1}w_5 \end{array}\right) \tag{5.3.13}$$

This is the key result of our first stage of elimination, which has removed the fast bulk variables.

5.3.1 Bulk enzyme elimination as quasi-steady state method

Before moving on to the second stage of also eliminating the subnetwork enzymes, we pause briefly to give a simpler form of our last result. While (5.3.13) gives a closed form for the effective L-matrix $\boldsymbol{L}_{\backslash \mathrm{e}} + \Delta\boldsymbol{L}_{\backslash \mathrm{e}}$ we obtain after eliminating the bulk enzymes, this form is not very intuitive. We show next that there is a much simpler statement of the result, namely that $\boldsymbol{L}_{\mathrm{eff}} = \boldsymbol{L}_{\backslash \mathrm{e}} + \Delta\boldsymbol{L}_{\backslash \mathrm{e}}$ can be obtained by treating all bulk enzymes as in quasi-steady state with the other molecular species.

To see this, we reinstate in the generic 3×3 block structure of (5.3.2) the specific notation for the linearised dynamics considered here, i.e.

$$\boldsymbol{L} = \left(\begin{array}{c|cc} m & w_1 & f_1 \\ \hline w_2 & w_3 & f_2 \\ w_4 & w_5 & f_3 \end{array}\right) = \left(\begin{array}{c|cc} \boldsymbol{L}^{\mathrm{S,S}} & \boldsymbol{L}^{\mathrm{S,b}} & \boldsymbol{L}^{\mathrm{S,e}} \\ \hline \boldsymbol{L}^{\mathrm{b,S}} & \boldsymbol{L}^{\mathrm{b,b}} & \boldsymbol{L}^{\mathrm{b,e}} \\ \boldsymbol{L}^{\mathrm{e,S}} & \boldsymbol{L}^{\mathrm{e,b}} & \boldsymbol{L}^{\mathrm{e,e}} \end{array}\right) \tag{5.3.14}$$

where as before S collects all subnetwork variables, i.e. subnetwork proteins s and subnetwork enzymes e′. The dynamics of the system can then be written as

$$\begin{aligned} \frac{\partial}{\partial t}\boldsymbol{\delta x}^{\mathrm{S^T}} &= \boldsymbol{\delta x}^{\mathrm{S^T}}\boldsymbol{L}^{\mathrm{S,S}} + \boldsymbol{\delta x}^{\mathrm{b^T}}\boldsymbol{L}^{\mathrm{b,S}} + \boldsymbol{\delta x}^{\mathrm{e^T}}\boldsymbol{L}^{\mathrm{e,S}} \\ \frac{\partial}{\partial t}\boldsymbol{\delta x}^{\mathrm{b^T}} &= \boldsymbol{\delta x}^{\mathrm{S^T}}\boldsymbol{L}^{\mathrm{S,b}} + \boldsymbol{\delta x}^{\mathrm{b^T}}\boldsymbol{L}^{\mathrm{b,b}} + \boldsymbol{\delta x}^{\mathrm{e^T}}\boldsymbol{L}^{\mathrm{e,b}} \\ \frac{\partial}{\partial t}\boldsymbol{\delta x}^{\mathrm{e^T}} &= \boldsymbol{\delta x}^{\mathrm{S^T}}\boldsymbol{L}^{\mathrm{S,e}} + \boldsymbol{\delta x}^{\mathrm{b^T}}\boldsymbol{L}^{\mathrm{b,e}} + \boldsymbol{\delta x}^{\mathrm{e^T}}\boldsymbol{L}^{\mathrm{e,e}} \end{aligned} \tag{5.3.15}$$

If we now impose a quasi-steady state condition for the bulk enzymes $\boldsymbol{\delta x}^{\mathrm{e}}$, this gives

$$\boldsymbol{\delta x}^{\mathrm{e^T}} = -\left(\boldsymbol{\delta x}^{\mathrm{S^T}}\boldsymbol{L}^{\mathrm{S,e}} + \boldsymbol{\delta x}^{\mathrm{b^T}}\boldsymbol{L}^{\mathrm{b,e}}\right)\boldsymbol{L}^{\mathrm{e,e}-1} \tag{5.3.16}$$

Substituting this back into the equations of motion for the subnetwork species and the

bulk proteins gives

$$\begin{aligned}\frac{\partial}{\partial t}\boldsymbol{\delta x}^{\mathrm{S^T}} &= \boldsymbol{\delta x}^{\mathrm{S^T}}\left(\boldsymbol{L}^{\mathrm{S,S}} - \boldsymbol{L}^{\mathrm{S,e}}\boldsymbol{L}^{\mathrm{e,e-1}}\boldsymbol{L}^{\mathrm{e,S}}\right) + \boldsymbol{\delta x}^{\mathrm{b^T}}\left(\boldsymbol{L}^{\mathrm{b,S}} - \boldsymbol{L}^{\mathrm{b,e}}\boldsymbol{L}^{\mathrm{e,e-1}}\boldsymbol{L}^{\mathrm{e,S}}\right)\\ \frac{\partial}{\partial t}\boldsymbol{\delta x}^{\mathrm{b^T}} &= \boldsymbol{\delta x}^{\mathrm{S^T}}\left(\boldsymbol{L}^{\mathrm{S,b}} - \boldsymbol{L}^{\mathrm{S,e}}\boldsymbol{L}^{\mathrm{e,e-1}}\boldsymbol{L}^{\mathrm{e,b}}\right) + \boldsymbol{\delta x}^{\mathrm{b^T}}\left(\boldsymbol{L}^{\mathrm{b,b}} - \boldsymbol{L}^{\mathrm{b,e}}\boldsymbol{L}^{\mathrm{e,e-1}}\boldsymbol{L}^{\mathrm{e,b}}\right)\end{aligned} \tag{5.3.17}$$

This is exactly the dynamics that is defined by the effective L-matrix $\boldsymbol{L}_{\mathrm{eff}} = \boldsymbol{L}_{\backslash \mathrm{e}} + \Delta\boldsymbol{L}_{\backslash \mathrm{e}}$ derived above (see (5.3.12) and (5.3.13)), hence proving our claim that this matrix can be constructed by imposing a quasi-steady state condition for the bulk enzymes.

5.3.2 Michaelis-Menten terms as effective unary reactions

The above bulk enzyme elimination can be carried out in closed form, and gives a simple result for the effective L-matrix. Looking at (5.3.1), each bulk enzyme can be eliminated by setting the time derivative of its concentration to zero. Here and throughout we assume that each enzyme only catalyses one reaction. (In the matrix formulation, this implies that the block $\boldsymbol{L}^{\mathrm{e,e}}$ is diagonal.) This gives for the equations of its substrate and product the following form:

$$\begin{aligned}\frac{\partial}{\partial t}\delta x_u &= -\lambda_{up}\delta x_u + \lambda_{pu}(y_p/y_u)\delta x_p + \ldots\\ \frac{\partial}{\partial t}\delta x_p &= -\lambda_{pu}\delta x_p + \lambda_{up}(y_u/y_p)\delta x_u + \ldots\end{aligned} \tag{5.3.18}$$

where

$$\begin{aligned}\lambda_{up} &= \frac{k^+_{ue,c}y_e\left(k^-_{c,pe} + k^+_{pe,c}y_p\right)}{k^-_{c,ue} + k^-_{c,pe} + k^+_{ue,c}y_u + k^+_{pe,c}y_p}\\ \lambda_{pu} &= \frac{k^+_{pe,c}y_e\left(k^-_{c,ue} + k^+_{ue,c}y_u\right)}{k^-_{c,ue} + k^-_{c,pe} + k^+_{ue,c}y_u + k^+_{pe,c}y_p}\end{aligned} \tag{5.3.19}$$

are the rates for effective unary reactions converting substrate to product and back, respectively. The factors of (y_p/y_u) in (5.3.18) arise because we are using dimensionless concentration variables.

The bulk enzyme elimination thus has the simple effect of replacing all bulk Michaelis-Menten reactions by unary conversion reactions with constant rates. From (5.2.11) one sees that these effective rates can be expressed directly in terms of the Michaelis-Menten parameters, as

$$\begin{aligned}\lambda_{up} &= \frac{V_{\mathrm{u}}/K_{\mathrm{u}} + (V_{\mathrm{u}} + V_{\mathrm{p}})/K_{\mathrm{u}}(y_p/K_{\mathrm{p}})}{(1 + y_u/K_{\mathrm{u}} + y_p/K_{\mathrm{p}})^2}\\ \lambda_{pu} &= \frac{V_{\mathrm{p}}/K_{\mathrm{p}} + (V_{\mathrm{u}} + V_{\mathrm{p}})/K_{\mathrm{p}}(y_u/K_{\mathrm{u}})}{(1 + y_u/K_{\mathrm{u}} + y_p/K_{\mathrm{p}})^2}\end{aligned} \tag{5.3.20}$$

Comparing with (5.1.14) shows that the rates are obtained simply by linearising the Michaelis-Menten reaction flux around the steady state concentrations of substrate and product. This is the closed-form procedure for bulk enzyme elimination we were after: it only requires as input the Michaelis-Menten parameters of the original network, and its steady state.

Note that the above discussion includes enzyme reactions both entirely in the bulk, or on the boundary of subnetwork and bulk (cf. Fig. 5.4b and c). The only difference between these two cases is that for the latter group of enzymes, the effective unary reactions we have derived are between a subnetwork boundary species and a bulk species and so will contribute to the rate matrix, while for enzyme reactions entirely in the bulk the effective reactions only affect the memory function.

5.3.3 Elimination of subnetwork enzymes

So far we have described how bulk enzymes can be eliminated, replacing them by effective unary conversion reactions. This allows the rate matrix and memory functions to be calculated from an effective L-matrix $\boldsymbol{L}_{\text{eff}} = \boldsymbol{L}_{\backslash \text{e}} + \Delta\boldsymbol{L}_{\backslash \text{e}}$. These quantities determine the projected equations of motion for the subnetwork proteins s and the subnetwork enzymes e$'$.

The second, final stage of the elimination procedure is now to eliminate the subnetwork enzymes. We claim that elimination of the bulk enzymes does not affect the equations of motion for the subnetwork enzymes, i.e. these do not acquire memory terms nor are the local-in-time terms changed. To see this, write the additional term (5.3.13) in the effective L-matrix as

$$\Delta\boldsymbol{L}_{\backslash \text{e}} = -\begin{pmatrix} \bar{f}_1 \\ \bar{f}_2 \end{pmatrix} (\bar{f}_3)^{-1} \begin{pmatrix} w_4 & w_5 \end{pmatrix} = -\begin{pmatrix} \boldsymbol{L}^{\text{S,e}} \\ \boldsymbol{L}^{\text{b,e}} \end{pmatrix} (\boldsymbol{L}^{\text{e,e}})^{-1} \begin{pmatrix} \boldsymbol{L}^{\text{e,S}} & \boldsymbol{L}^{\text{e,b}} \end{pmatrix} \tag{5.3.21}$$

"Unpacking" S into s and e$'$ gives

$$\Delta\boldsymbol{L}_{\backslash \text{e}} = -\begin{pmatrix} \boldsymbol{L}^{\text{s,e}} \\ \boldsymbol{L}^{\text{e}',\text{e}} \\ \boldsymbol{L}^{\text{b,e}} \end{pmatrix} (\boldsymbol{L}^{\text{e,e}})^{-1} \begin{pmatrix} \boldsymbol{L}^{\text{e,s}} & \boldsymbol{L}^{\text{e,e}'} & \boldsymbol{L}^{\text{e,b}} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{L}^{\text{s,e}} \\ 0 \\ \boldsymbol{L}^{\text{b,e}} \end{pmatrix} (\boldsymbol{L}^{\text{e,e}})^{-1} \begin{pmatrix} \boldsymbol{L}^{\text{e,s}} & 0 & \boldsymbol{L}^{\text{e,b}} \end{pmatrix} \tag{5.3.22}$$

This shows that both the e′ block row and block column of $\Delta \boldsymbol{L}_{\backslash \mathrm{e}}$ vanish: intuitively, the effective unary reactions from bulk enzymes do not couple to subnetwork enzymes. Specifically, $\Delta \boldsymbol{L}_{\backslash \mathrm{e}}^{\mathrm{b,e'}} = 0$, and since also in the original dynamics the subnetwork enzymes do not interact with bulk species ($\boldsymbol{L}^{\mathrm{b,e'}} = 0$), one has $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{b,e'}} = 0$. Looking at the general formula (5.2.5) for the memory function, read in terms of the effective L-matrix, then confirms that the equations for the e′ species do not contain memory terms. This makes sense: the subnetwork enzymes are not boundary species to start with, and this is not changed by the effective unary reactions from bulk enzymes.

To sum up so far, the projected equation for each subnetwork enzyme looks exactly as the original equation in (5.2.12). Because of the fast rate γ in this, in the limit $\gamma \to \infty$ each subnetwork enzyme will be in quasi-steady state with its substrate and product. Substituting the quasi-steady state enzyme concentration into the equations for substrate and product then gives again effective unary conversion reactions, with rates as given in (5.3.20) for the case of bulk enzymes.

5.3.4 Summary of enzyme elimination procedure for linearised dynamics

The final procedure we have arrived at for constructing projected equations for reaction systems with Michaelis-Menten terms, within linearised dynamics, is remarkably simple: replace each Michaelis-Menten term, whether in the subnetwork, the bulk or on the boundary, by its linearisation around the steady state. This gives effective rates for unary conversion reactions among each substrate-product pair (see (5.3.19)).

5.4 Nonlinear Dynamics

We want to extend the above approach to eliminate enzymes from the projected equations for the full nonlinear dynamics. Directly transplating the results from the linearised dynamics is not possible, however: if we use the quasi-steady state assumption for the enzymes as in Section 5.3, then we get back the full Michaelis-Menten nonlinearities. As these go beyond second order in δx, they cannot be used directly in our construction of the projected equations, which starts from reaction equations with only linear

and quadratic terms, as appropriate for a mass action description of unary and binary reactions. We want to find a solution that is compatible with a mass action system of equations.

We take as our starting point the nonlinear L-matrix, as shown in equation (5.2.2), but subdivide this into smaller blocks below to single out contributions from enzymes. Focussing though for now just on the distinction between linear and quadratic observables, we have two new kinds of entries. Firstly, mixed linear-quadratic elements as contained in e.g. $\boldsymbol{L}^{\mathrm{ss,s}}$: these are coefficients of quadratic terms in equations of motion for the concentrations (linear observables), so can be read off directly from the mass action equations. The quadratic-quadratic elements as in $\boldsymbol{L}^{\mathrm{ss,ss}}$ are coefficients from equations of motion of concentration products. For a generic product $\delta x_{ij} \equiv \delta x_i \delta x_j$ these are of the form

$$\frac{\partial}{\partial t}\delta x_i \delta x_j = \delta x_j \frac{\partial}{\partial t}\delta x_i + \delta x_i \frac{\partial}{\partial t}\delta x_j \tag{5.4.1}$$

Because we are only considering terms up to quadratic order on the r.h.s., we need to insert only the linearised equations of motion for $(\partial \delta x_i/\partial t)$ and $(\partial \delta x_j/\partial t)$. All quadratic-quadratic elements of the L-matrix are therefore "inherited" from the linearised dynamics. In particular, the above structure of the equations of motion for quadratic observables means that the time evolution of any product containing at least one enzyme factor will contain fast terms.

For the purpose of eliminating the fast degrees of freedom, the nonlinear L-matrix can be split into four blocks mirroring the structure of the linear L-matrix in equation (5.3.2), viz.

$$\boldsymbol{L} = \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{S,S}} & \boldsymbol{L}^{\mathrm{S,B}} \\ \hline \boldsymbol{L}^{\mathrm{B,S}} & \boldsymbol{L}^{\mathrm{B,B}} \end{array}\right) = \left(\begin{array}{cc|cc} \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{e}}'} & \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{e}}} \\ \boldsymbol{L}^{\tilde{\mathrm{e}}',\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{e}}',\tilde{\mathrm{e}}'} & \boldsymbol{L}^{\tilde{\mathrm{e}}',\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{e}}',\tilde{\mathrm{e}}} \\ \hline \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{e}}'} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{e}}} \\ \boldsymbol{L}^{\tilde{\mathrm{e}},\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{e}},\tilde{\mathrm{e}}'} & \boldsymbol{L}^{\tilde{\mathrm{e}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{e}},\tilde{\mathrm{e}}} \end{array}\right) \tag{5.4.2}$$

The blocks are defined so that $\tilde{\mathrm{s}}$ contains the observables s and ss which have no bulk or fast factors, $\tilde{\mathrm{b}}$ contains the observables b, sb and bb which have at least one bulk factor but no fast factors, $\tilde{\mathrm{e}}$ consists of e, se, be, be$'$, ee and ee$'$ which contain at least one bulk factor and one fast factor - where the fast and bulk factor can be identical - and $\tilde{\mathrm{e}}'$ contains e$'$, se$'$ and e$'$e$'$ where there are no bulk factors but at least one fast factor. Therefore the subnetwork block S consists of slow and fast blocks in the form of $\tilde{\mathrm{s}}$ and

$\tilde{e}'$ respectively and similarly the bulk block B contains slow and fast contributions from $\tilde{b}$ and $\tilde{e}$.

The block structure of the L-matrices for the linearised and nonlinear dynamics is therefore the same; however, there are some differences. The block $\tilde{e}\tilde{e}$ contains some slow as well as fast entries, but the slow entries can be neglected in comparison in the large γ limit. The $\tilde{e}\tilde{e}'$ block is not zero as in the linear case due to the fact that the equation of motion for se' involves products of the form be' and ee'; importantly for our reasoning below these terms are slow, however, because of the time evolution of s in the se' products. Similarly the $\tilde{e}'\tilde{e}$ block is not zero because the equation of motion for be' involves slow se' contributions; also in the equation for ee' there are fast contributions from se'. The blocks $\tilde{b}\tilde{e}'$ and $\tilde{e}'\tilde{b}$ remain zero, on the other hand.

We can go back to the same 3×3 structure for the L-matrix as in the linear case, by partitioning into blocks S ($\tilde{s}$ and $\tilde{e}'$), $\tilde{b}$ and $\tilde{e}$. This 3×3 matrix can then be split into fast and slow blocks as in equation (5.3.2):

$$\boldsymbol{L} = \left(\begin{array}{c:cc} \boldsymbol{L}^{\mathrm{S,S}} & \boldsymbol{L}^{\mathrm{S},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\mathrm{S},\tilde{\mathrm{e}}} \\ \hdashline \boldsymbol{L}^{\tilde{\mathrm{b}},\mathrm{S}} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{e}}} \\ \boldsymbol{L}^{\tilde{\mathrm{e}},\mathrm{S}} & \boldsymbol{L}^{\tilde{\mathrm{e}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{e}},\tilde{\mathrm{e}}} \end{array}\right) = \left(\begin{array}{c:cc} m & w_1 & f_1 \\ \hdashline w_2 & w_3 & f_2 \\ w_4 & w_5 & f_3 \end{array}\right) \tag{5.4.3}$$

Using the method of Sec. 5.3 we can then find the slow and fast parts of the memory expressed in terms of these blocks, using exactly the same formulae as shown for the linearised dynamics in equation (5.3.8). As before the result can be thought of as arising from a modified L-matrix where all the fast bulk species and products contained in $\tilde{e}$ are eliminated:

$$\boldsymbol{L}_{\mathrm{eff}} = \left(\begin{array}{c:c} m - \bar{f}_1(\bar{f}_3)^{-1}w_4 & w_1 - \bar{f}_1(\bar{f}_3)^{-1}w_5 \\ \hdashline w_2 - \bar{f}_2(\bar{f}_3)^{-1}w_4 & w_3 - \bar{f}_2(\bar{f}_3)^{-1}w_5 \end{array}\right) \tag{5.4.4}$$

The form of $\boldsymbol{L}_{\mathrm{eff}}$ can be derived by eliminating the fast observables in $\tilde{e}$ directly, by assuming that they are in steady state with respect to the fast contributions from their equations of motion. We turn next to the task of actually carrying out this elimination, which is more involved than in the linear case.

5.4.1 Elimination of fast bulk variables

We can simplify matters somewhat by noting that to find the correct rate matrix and memory function with enzymes eliminated, we require the equations of motion – and hence the relevant columns of $\boldsymbol{L}_{\text{eff}}$ – for the bulk observables $\tilde{\text{b}}$ and the *linear* subnetwork observables s and e′ with enzymes eliminated. We only need the linear observables s and e′ because to calculate the rate matrix and memory function we do not need blocks of the form $\boldsymbol{L}^{\cdot,\text{ss}}$, as explained in Sec. 5.2 above. A further simplification comes from the fact that the original equations of motion for $\tilde{\text{b}}$, s and e′ do not depend on ee or ee′ and therefore we do not need to consider these observables further and can focus on how to eliminate the remaining fast observables se, be, be′ and e.

We consider first the product observables se, be and be′. The equation of motion for a generic observable of type se, i.e. a product of the concentration of a subnetwork species with that of an enzyme *se*, reads

$$\frac{\partial}{\partial t}\delta x_{se} = \frac{\partial}{\partial t}\delta x_s \delta x_e = \delta x_s \frac{\partial}{\partial t}\delta x_e + \delta x_e \frac{\partial}{\partial t}\delta x_s = \delta x_s \frac{\partial}{\partial t}\delta x_e + \text{slow} \tag{5.4.5}$$

where we have used that $(\partial/\partial t)\delta x_s$ only contains slow terms. In finding the solution of the quasi-steady state condition $(\partial/\partial t)\delta x_{se} = 0$, these slow terms can be neglected compared to the fast terms from $(\partial/\partial t)\delta x_e$. Writing the latter in the form

$$\frac{\partial}{\partial t}\delta x_e = A\delta x_u + B\delta x_p - C\delta x_e \tag{5.4.6}$$

This allows us to write leading (fast) terms in the equation for the subnetwork-enzyme product as

$$\frac{\partial}{\partial t}\delta x_{se} = \delta x_s \frac{\partial}{\partial t}\delta x_e = A\,\delta x_{su} + B\,\delta x_{sp} - C\,\delta x_{se} \tag{5.4.7}$$

Setting this to zero shows that the quasi-steady state solution is

$$\delta x_{se} = (A/C)\delta x_{su} + (B/C)\delta x_{sp} \tag{5.4.8}$$

Comparing with the (linear) quasi-steady state solution for the enzyme concentration itself, which is $\delta x_e = (A/C)\delta x_u + (B/C)\delta x_p$, we arrive at a simple product elimination rule: products of the form se are eliminated by using the linear elimination of the enzyme, multiplying by a factor of δx_s, and then identifying $\delta x_{su} = \delta x_s \delta x_u$ and $\delta x_{sp} = \delta x_s \delta x_p$. It is straightforward to check that the same rule applies to the elimination of observables of type be and be′.

The only remaining fast observables that we need to eliminate are the linear bulk enzyme concentrations. Their equations of motion from (5.2.12) and (5.2.13) are

$$\frac{\partial}{\partial t}\delta x_e = -(\gamma/y_e)\left[k^-_{c,ue}y_e\delta x_e + k^+_{ue,c}y_u y_e(\delta x_u + \delta x_e + \delta x_{ue}) \right. \\ \left. + k^-_{c,pe}y_e\delta x_e + k^+_{pe,c}y_p y_e(\delta x_p + \delta x_e + \delta x_{pe})\right] \quad (5.4.9)$$

These contain product variables of the form se and be, which can now be eliminated using the method above, giving expressions in the form of equation (5.4.8). Substituting these and solving the quasi-steady state condition $(\partial/\partial t)\delta x_e = 0$ then gives

$$\delta x_e = -\frac{k^+_{ue,c}y_u}{(k^-_{c,ue} + k^+_{ue,c}y_u + k^-_{c,pe} + k^+_{pe,c}y_p)^2}\Big((k^-_{c,ue} + k^+_{ue,c}y_u + k^-_{c,pe} + k^+_{pe,c}y_p)\delta x_u \\ + k^+_{ue,c}y_u\delta x_{uu} + k^+_{pe,c}y_p\delta x_{up}\Big) \\ - \frac{k^+_{pe,c}y_p}{(k^-_{c,ue} + k^+_{ue,c}y_u + k^-_{c,pe} + k^+_{pe,c}y_p)^2}\Big((k^-_{c,ue} + k^+_{ue,c}y_u + k^-_{c,pe} + k^+_{pe,c}y_p)\delta x_p \\ + k^+_{ue,c}y_u\delta x_{up} + k^+_{pe,c}y_p\delta x_{pp}\Big) \quad (5.4.10)$$

We can now compare to the standard Michaelis-Menten elimination of the bulk enzyme, which treats products like δx_{ue} not as separate observables but identifies them with $\delta x_u\delta x_e$ and then solves $(\partial/\partial t)\delta x_e = 0$. It is straightforward to check that our above elimination formula is just this Michaelis-Menten result expanded to quadratic order. We will therefore call this result "quadratic quasi-steady state elimination".

The result of the first stage of elimination is therefore that we can construct equations of motion for s, e′, b, sb and bb by quadratic quasi-steady state elimination of the bulk enzymes. The full quadratic elimination is not needed for all observables as the equations of motion for sb and bb only contain quadratic observables: in these we use linear enzyme elimination to replace products as explained above.

We can now write down explicitly what the effective contributions to the equations of motion for a substrate and product are. One starts from (5.2.12) and (5.2.13) again,

$$\frac{\partial}{\partial t}\delta x_u = -(1/y_u)\left[k^-_{c,ue}y_e\delta x_e + k^+_{ue,c}y_u y_e(\delta x_u + \delta x_e + \delta x_{ue})\right] + \ldots \quad (5.4.11)$$

and substitutes in the elimination formulae for δx_e and δx_{ue}. After a little algebra, the effective equation of motion for the substrate, and analogously the product, can be

written in terms of unary reactions, as was the case for the linearised dynamics:

$$\begin{aligned}\frac{\partial}{\partial t}\delta x_u &= -\hat{\lambda}_{up}\delta x_u + \hat{\lambda}_{pu}(y_p/y_u)\delta x_p + \dots \\ \frac{\partial}{\partial t}\delta x_p &= -\hat{\lambda}_{pu}\delta x_p + \hat{\lambda}_{up}(y_u/y_p)\delta x_u + \dots\end{aligned} \tag{5.4.12}$$

The difference is that the reaction rates $\hat{\lambda}_{up}$ and $\hat{\lambda}_{pu}$ are now linearly dependent on substrate and product concentrations. In terms of the (constant) reaction rates λ_{pu} and λ_{up} defined in (5.3.20), we can write this concentration dependence in the simple form

$$\begin{aligned}\hat{\lambda}_{up} &= \lambda_{up}\left(1 - \frac{\delta x_u y_u/K_\mathrm{u} + \delta x_p y_p/K_\mathrm{p}}{1 + y_u/K_\mathrm{u} + y_p/K_\mathrm{p}}\right) \\ \hat{\lambda}_{pu} &= \lambda_{pu}\left(1 - \frac{\delta x_u y_u/K_\mathrm{u} + \delta x_p y_p/K_\mathrm{p}}{1 + y_u/K_\mathrm{u} + y_p/K_\mathrm{p}}\right)\end{aligned} \tag{5.4.13}$$

Representing every Michaelis-Menten term in the bulk or on the boundary in this form, we thus obtain a set of equations from which all bulk enzymes have been eliminated. The coefficients in these equations then define the effective L-matrix $\boldsymbol{L}_\mathrm{eff}$ (or more precisely those columns of it that we use to obtain the rate and memory matrix.

5.4.2 Elimination of fast subnetwork observables

The result of the first stage of elimination is a rate matrix and memory matrix for the projected equations of the subnetwork variables s and e′. The second and final stage in the elimination of fast observables is now to remove the subnetwork enzymes e′.

This second stage is relatively simple because the projected equations of motion for e′ observables are in fact just the original mass action equations. This is because the equations of motion for e′ observables only contain fast contributions from e′ and se′, so do not couple to any variables that were eliminated in the first stage. Furthermore, e′ observables are interior subnetwork species and do not couple to the slow bulk variables. Therefore their equations of motion cannot acquire memory terms. Eliminating subnetwork enzymes is then trivial: in the limit $\gamma \to \infty$ the quasi-steady state assumption will become exact for them. Solving this gives exactly the standard Michaelis-Menten expression for the enzyme concentrations.

The final question is then where the subnetwork enzymes e′ feature in the projected equations for slow subnetwork species s. They appear in the rate matrix and substituting them there simply produces the usual Michaelis-Menten terms; note though that these

appear in their full nonlinear form rather than being expanded to second order as for bulk enzymes.

In the memory function e' cannot appear linearly as subnetwork enzymes are not on the boundary. Such enzymes could then only appear in the memory via se' terms. For this to occur would require nonzero entries in the blocks $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{b}}$, $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{sb}}$ or $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{bb}}$, which are the relevant pieces of the leftmost factor $L^{\mathrm{S,B}}$ in (5.2.6). The block $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{b}}$ contains the nonlinear se' contributions in the equation of motion for a bulk observable; however such an equation cannot involve any se' products because subnetwork enzymes do not interact with the bulk. Therefore there are no contributions to $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{b}}$. Similarly $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{bb}}$ must be zero as from the product rule of differentiation in (5.4.1) there would have to be a shared species in the first and second product index in order to obtain a nonzero contribution. For the block $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{sb}}$ to have nonzero entries there must be a shared index, which in this case must be s. This means that nonzero elements could come only from the linearised equation of motion for the b observable; however, there are no contributions from e' in the equations of motion for bulk observables and therefore $\boldsymbol{L}_{\mathrm{eff}}^{\mathrm{se}',\mathrm{sb}}$ must also be zero. In summary, this means that there are no contributions from subnetwork enzymes to any memory function.

5.4.3 Summary of enzyme elimination procedure for nonlinear dynamics

The final procedure we have arrived at for constructing projected equations for reaction systems with Michaelis-Menten terms, for nonlinear dynamics, can be split into two simple steps. The first is to construct the reduced L-matrix $\boldsymbol{L}_{\mathrm{eff}}$ by expanding all the Michaelis-Menten terms to second order around the steady state. From this matrix we are then able to calculate the rate matrix and memory function. Once this is done, we simply reinstate the full nonlinear form of the Michaelis-Menten terms from enzymes in the subnetwork.

5.5 Numerical Comparisons

The EGFR network model as shown in Fig. 5.6 has two subnetwork Michaelis-Menten reactions and one bulk Michaelis-Menten reaction. Here we choose the bulk to be the protein Shc and all complexes that include Shc as explained in Chapter 4. In this section we want to compare two versions of the projected equations: the ones obtained by adding enzyme species explicitly and finite but large enzyme reaction rates, and the ones we get by eliminating enzymes explicitly as explained above. We will also compare both projected descriptions with the dynamics of the full EGFR network [13], i.e. tracking explicitly the bulk degrees of freedom, to see which represents the true time courses better. The bulk is assumed to be at steady state initially and therefore the random force is set to zero and the subnetwork initial conditions are chosen to maximise nonlinear effects subject to the constraint that all conserved quantities are zero at steady state. We again use Δ as defined in (5.1.22) to measure the difference between time courses where $T = 150$.

5.5.1 Explicit enzyme reactions

For the projected equations with enzymes represented explicitly, we need to convert the three (irreversible) Michaelis-Menten reactions in the EGFR network model [13].

$$\begin{aligned} \mathrm{RP} &\rightleftharpoons \mathrm{R}_2 \\ \mathrm{PLC}\gamma\mathrm{P} &\rightleftharpoons \mathrm{PLC}\gamma \\ \mathrm{ShP} &\rightleftharpoons \mathrm{Shc} \end{aligned} \tag{5.5.1}$$

into mass action form. For each we use the relevant K_m and V_max values together with a suitable value of the fast rate γ to create a set of mass action reaction rates. The relevant mass action terms are then added to the equations for substrate and product, and we add a mass action equation for the concentrations of enzyme and enzyme complex, as in (5.1.2). The steady states of the enzyme and enzyme complex, which we require for the construction of the projected equations, can then be found by solving the relevant equations for the substrate, enzyme and enzyme complex. We note that to solve the full system of equations, the steady state values of the enzyme-substrate complexes must be added to the relevant conservation laws to ensure that the correct steady state is

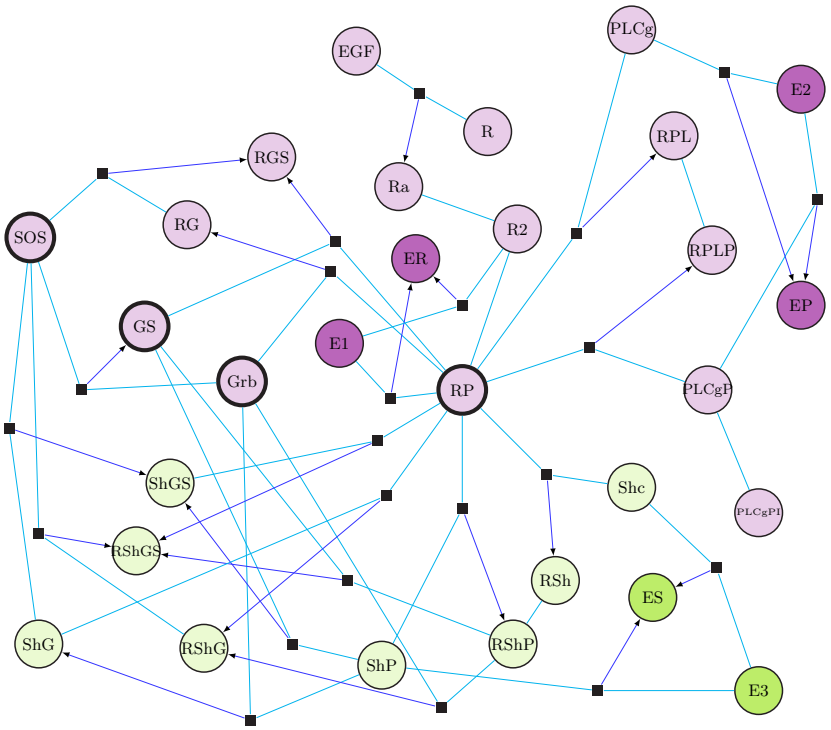


Figure 5.6: Factor graph of EGFR network as described in Kholodenko et al. [13] adapted to include enzyme reactions. Three added enzyme reactions (highlighted) with enzymes denoted E1-3 and enzyme-substrate complexes "enzyme-R" (denoted ER), "enzyme-PLCγ" (denoted EP), "enzyme-Shc" (denoted ES) capture the Michaelis-Menten contributions to the dynamics.

reached, with the same concentrations as in the Michaelis-Menten description. From this explicit expanded mass action system we can then construct projected equations, either linearised or nonlinear, in the standard manner explained in Sec. 3.2.

5.5.2 Enzyme elimination in linearised dynamics

The elimination procedure for the linearised dynamics as described in Section 5.3 is applied before we calculate the projected equations. We write the enzyme reactions as a conformational change between substrate and product and construct from the resulting $\boldsymbol{L}_{\text{eff}}$ a set of projected equations for the subnetwork observables; this construction now no longer makes explicit reference to enzymes. Fig. 5.7 compares the linearised projected equations with enzymes eliminated to the projected equations with explicit enzymes, as a function of the fast enzyme reaction scale γ used in the explicit case. We can see that

as the size of γ increases the mass action system with explicit enzymes approaches the system with enzymes eliminated. This is exactly what we expect, because we constructed the method of enzyme elimination to be exact in the large γ limit.

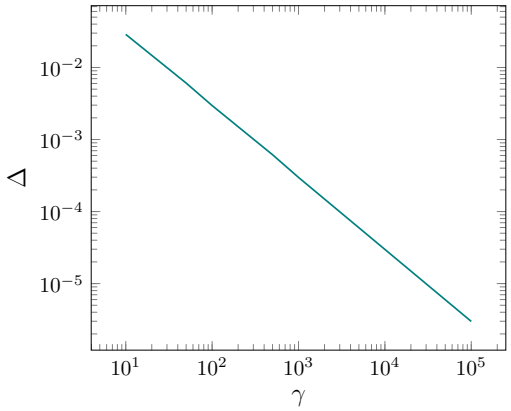


Figure 5.7: Errors between linearised projected equations with explicit enzymes and with enzymes eliminated, as a function of the fast enzyme rate γ.

5.5.3 Enzyme elimination in nonlinear dynamics

We apply the elimination method from Section 5.4 to the full nonlinear reaction system. We compare the resulting projected equations where enzymes have been eliminated to the (nonlinear) projected equations derived from the expanded mass action systems where the enzymes are represented explicitly. Figure 5.8 shows that the deviation between the time courses predicted by the two sets of equations again decreases towards zero with increasing γ as it should. We now need a substantial value of γ, of order $10^4 \mathrm{s}^{-1}$, to make the deviation quantitatively small. We also note that computation times required to integrate the system of projected equations derived by enzyme elimination are shorter by roughly a factor of 2 than those for the case where enzymes are represented explicitly; this is due to the appearance of "stiff" terms in the latter case, which the enzyme elimination avoids. Fig. 5.9 compares the full Michaelis-Menten reaction equations to the projected equations with enzymes eliminated and with enzymes in mass action form. We can see that to model Michaelis-Menten reactions we should not be using finite γ as this is the main cause of errors.

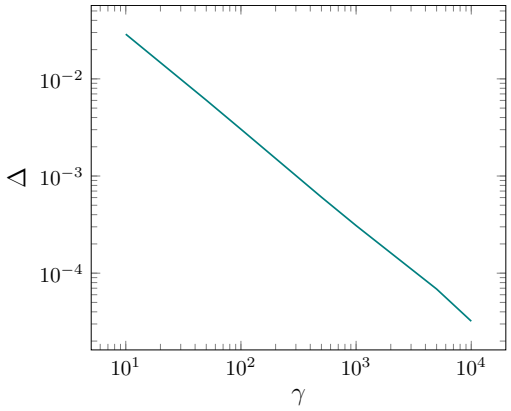


Figure 5.8: Deviation between time courses predicted by nonlinear projected equations with explicit enzymes and with enzymes eliminated, as a function of enzyme rate scale γ.

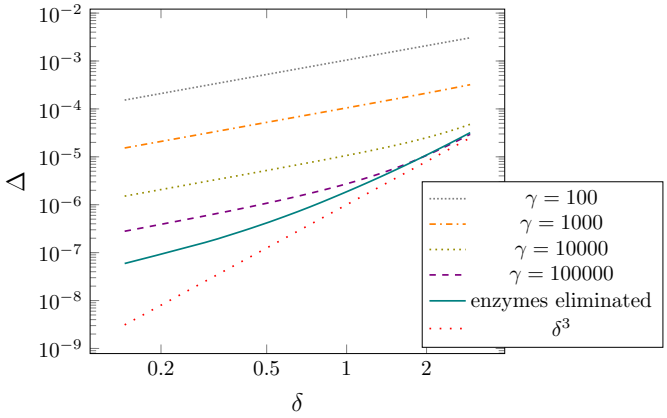


Figure 5.9: Plot of approximation error vs initial deviation from steady state in log-log representation, for the nonlinear projected equations with enzymes eliminated and projected equations with enzyme rates dependent on different values of γ. Red dotted line is proportional to δ^3 to demonstrate that the approximation error of the projected equations grows cubically in δ.

5.6 Discussion

We have considered the problem of describing subnetwork dynamics in protein interaction networks. The projection approach we have previously developed gives accurate results in this regard, but can be applied directly only to systems with unary and binary reactions. Our aim in this chapter was to extend it to systems involving enzymatic reactions represented as Michaelis-Menten terms.

We started with a careful discussion of the conditions under which enzyme reactions represented in mass action form become equivalent to Michaelis-Menten kinetics. The result is a convenient scaling, of fast enzyme reaction rates and simultaneously low enzyme concentrations, that exactly reproduces Michaelis-Menten equations in the limit where the relevant fast rate parameter γ grows large.

Applying this construction, one can map a protein interaction network with Michaelis-Menten reactions to one with only unary and binary reactions; the main task is then to understand what limit is approached in the projected description when $\gamma \to \infty$. By analysing what the fast enzyme degrees of freedom feature in the rate matrix and memory functions that define the projected equations, we showed that it is possible construct the projected equations directly in the large γ-limit, both linearised and nonlinear dynamics, using a quasi-steady state elimination. This gives us effective unary reaction contributions to represent the Michaelis-Menten dynamics, with concentration-dependent reaction rates in the nonlinear case, and so allows one to constuct the projected equations without ever introducing enzymes explicitly.

The resulting method significantly widens the range of biochemical reaction systems to which the projection approach can be applied; as one example we showed an application to a subnetwork of the EGFR reaction network of Kholodenko et al. [13]. Our general approach to constructing projected equations for networks of nonlinear reaction equations should be more widely applicable still, and we hope in future work to consider e.g. Michaelis-Menten reactions with inhibition and Hill equations [51].

Chapter 6

Perturbations in protein subnetworks

To gain a better understanding of biochemical networks it is often necessary to use models that can incorporate experimental data. Therefore we need models that can capture the response of cells to perturbation. One of the most prevalent methods used to understand linear response is response analysis [64–66]. This is used to analyse the responses by perturbing each species in turn and then calculating response coefficients for all the species in the system with respect to the perturbed species. I now look at how the projection method can be used to analyse the response to perturbations.

In Chapter 3, I applied the Zwanzig-Mori projection method [27] to the problem of understanding subnetwork dynamics of protein interaction networks. Here I extend the approach for unary and binary protein interactions described in Chapter 3 to include creation and destruction rates that represent gene regulation. One of the main results in Chapter 3 is that there are memory effects describing the strength with which past subnetwork concentration values affect the present rate of change. These memory effects only act on the boundary species. I now consider the effects on the subnetwork found by perturbations of bulk species. One of our primary aims will be to understand whether the response to such perturbations has a similar boundary structure.

In Section 6.1, I summarise our method for obtaining projected equations for systems of unary and binary reactions and how the approach can be extended to include cre-

ation and destruction terms that represent gene regulation. Section 6.2 shows how we calculate the projected equations to linear order in the perturbation, which then allows one to describe linear response behaviour. We focus our analysis on the effects of bulk perturbations on the subnetwork and we show that the overall response can be split into an initial propagation of the perturbation from the perturbed bulk species to the boundary of the subnetwork, and a response of the subnetwork to this boundary perturbation. Both these terms can also be split into transient and persistent contributions depending on whether or not they decay in time. I illustrate this with a small example in Sec. 6.2.4 and then apply the formalism to the EFGR network in Sec. 6.2.5. In Section 6.2.6, I study briefly the connection between the memory function and response functions. Finally, in Sec. 6.3, I look at how the projection method can be extended to account for nonlinear response and show that this is necessary to obtain qualitatively accurate results for larger perturbations.

6.1 Methods

The aim of the projection method generally is to provide a description of the dynamics of a protein interaction subnetwork embedded in a bulk network. Specifically, the method can be used to obtain equations for the time evolution of the protein concentrations in a chosen subnetwork. Full details are explained in Chapter 3.

6.1.1 Extending the projection method

To understand the effects of perturbations we extend the list of possible protein reactions to include creation and destruction terms which represent gene regulation. Let θ_i be the rate of creation of a species i and γ_i be the rate of degradation of species i. The

deterministic reaction equations (3.1.1) are then written

$$\begin{aligned}\frac{\partial}{\partial t}\delta x_i = &\sum_{j(\neq i),l}\left[k^-_{l,ij}\delta x_l - k^+_{ij,l}(y_j\delta x_i + y_i\delta x_j + \delta x_i\delta x_j)\right] \\ &+ \frac{1}{2}\sum_{j\neq l}\left[k^+_{jl,i}(y_l\delta x_j + y_j\delta x_l + \delta x_j\delta x_l) - k^-_{i,jl}\delta x_i\right] \\ &+ \sum_{l}\left[2k^-_{l,ii}\delta x_l - k^+_{ii,l}(2y_i\delta x_i + \delta x_i\delta x_i)\right] \\ &+ \sum_{j}\left[\frac{1}{2}k^+_{jj,i}(2y_j\delta x_j + \delta x_j\delta x_j) - k^-_{i,jj}\delta x_i\right] \\ &+ \sum_{j}(\lambda_{ji}\delta x_j - \lambda_{ij}\delta x_i) + [\theta_i - \gamma_i(y_i + \delta x_i)]\end{aligned} \tag{6.1.1}$$

where $k^+_{ij,l}$ is the rate of formation of complex l from proteins i and j. The rate of the corresponding complex dissociation is denoted $k^-_{l,ij}$. The reaction rate λ_{ij} represents the rate of species i changing into species j.

The extra contribution to the reaction equations representing a perturbation is

$$[\theta_i - \gamma_i(y_i + \delta x_i)] = (h_i - \gamma_i\delta x_i) \tag{6.1.2}$$

where $h_i = \theta_i - \gamma_i y_i$. Protein reactions typically have a timescale of seconds whereas gene regulation typically occurs on a slower timescale [67, 68]. Therefore the values for θ_i and γ_i will generally be smaller than the other reaction rates.

There are other methods of modelling gene regulation. We have mostly followed the approach in [66]; however they assume that the rates θ_i and γ_i are themselves fast (corresponding to typical proteomic timescales of the order of seconds) and that the slower nature of gene regulation is represented in separate slow evolution equations for the θ_i. Barenco et al. [69] use a set of differential equations to model the gene transcript concentration. As well as creation and degratation rates they have a time dependent term representing the transcription. There are also stochastic models of gene regulation, for example, Ozbudak et al. [70] and Berg [71] use Langevin equations to model the dynamics of mRNA concentration.

We then use the reaction equations (6.1.1) to obtain a modified adjoint Fokker-Planck

operator that contains terms relating to γ_i and h_i.

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_0 + \delta\mathcal{L} \\
&= \sum_{i,j,l:i\neq j} \left[k^-_{l,ij}\delta x_l - k^+_{ij,l}(y_j\delta x_i + y_i\delta x_j + \delta x_i\delta x_j)\right]\frac{\partial}{\partial\delta x_i} \\
&\quad + \frac{1}{2}\sum_{i,j,l:j\neq l} \left[k^+_{jl,i}(y_l\delta x_j + y_j\delta x_l + \delta x_j\delta x_l) - k^-_{i,jl}\delta x_i\right]\frac{\partial}{\partial\delta x_i} \\
&\quad + \sum_{i,j}\Bigg\{\left[2k^-_{j,ii}\delta x_j - k^+_{ii,j}(2y_i\delta x_i + \delta x_i\delta x_i)\right] \\
&\quad + \frac{1}{2}\left[k^+_{jj,i}(2y_j\delta x_j + \delta x_j\delta x_j) - k^-_{i,jj}\delta x_i\right]\Bigg\}\frac{\partial}{\partial\delta x_i} \\
&\quad + \sum_{i,j}\left(\lambda_{ji}\delta x_j - \lambda_{ij}\delta x_i\right)\frac{\partial}{\partial\delta x_i} + \frac{\epsilon}{2}\sum_{i,j}\left(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\right)_{ij}\frac{\partial^2}{\partial\delta x_i\,\partial\delta x_j} + \sum_i\left(h_i - \gamma_i\delta x_i\right)\frac{\partial}{\partial\delta x_i}
\end{aligned} \tag{6.1.3}$$

We then define $\mathcal{L}_0$ as the original adjoint Fokker-Planck operator (3.1.4) that contained only unary and binary reactions together with the contribution from the destruction rates $-\gamma_i\delta x_i(\partial/\partial\delta x_i)$. The additional term $\delta\mathcal{L}$ is defined as

$$\delta\mathcal{L} = \sum_i h_i\frac{\partial}{\partial\delta x_i} \tag{6.1.4}$$

If $\boldsymbol{h}$ is zero then $\mathcal{L}_0$ represents the effects of both creation and destruction rates being present, but in exactly the ratio that is required in order not to affect the steady state. The new adjoint Fokker-Planck operator (6.1.3) will then be used in the construction of the projected equations.

To simplify the calculations for the rest of this work we will be using dimensionless variables $\delta\tilde{x}_i = \delta x_i/y_i$ and we have then dropped the tildes from the dimensionless variables.

6.1.1.1 Subnetwork perturbations

We first briefly consider the results of perturbing a subnetwork protein. The only effects will be in the rate matrix, from the destruction term γ, and the random force, from the perturbation term h. We are able to use the projected equations to analyse the timecourse of the reactions; however there will be no change to the memory functions. We therefore focus our analysis on the effects of perturbing bulk species.

6.2 Linear Response

For simplicity we first consider linear response around the steady state. Here we are considering linearised dynamics so that from $\mathcal{L}$ defined in (6.1.3) we retain only the terms that are linear in $\boldsymbol{\delta x}$. Correspondingly we choose as the set of observables to project onto simply the concentration deviations δx_i in the subnetwork.

6.2.1 Projected equations with gene regulation

We want to look at the effect on the subnetwork dynamics of regulation driving a change in concentration of a bulk species. The memory function $\boldsymbol{M}(t)$ and random force $\boldsymbol{r}(t)$ both contain the exponential operator $e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t}$, where the adjoint Fokker-Planck operator $\mathcal{L}$ now has an extra contribution from the perturbation. A small perturbation of an operator exponential can be written to linear order in the form [72]

$$e^{\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\mathcal{Q}t} = e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t} + \int_0^t dt'\; e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')} \tag{6.2.1}$$

This is essentially the first term in the Dyson series in quantum mechanics and the identity with the full L rather than L_0 in the rightmost exponential is exact and can be verified by differentiation.

The construction of the memory function for the projected equations is given in equation (5.2.5). Applying the relevant operators to a set of subnetwork species δx_i and using the exponential expansion (6.2.1) we find that

$$\begin{aligned}&\mathcal{P}(\mathcal{L}_0+\delta\mathcal{L})\mathcal{Q}e^{\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\mathcal{Q}t}\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\delta x_i = \\ &\quad\mathcal{P}(\mathcal{L}_0+\delta\mathcal{L})\mathcal{Q}\left(e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t} + \int_0^t dt'\; e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')}\right)\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\delta x_i\end{aligned} \tag{6.2.2}$$

We now show that the various occurrences of $\delta\mathcal{L}$ here only give vanishing contributions. For the rightmost one, because we are considering bulk perturbations but the observables δx_i we project are in the subnetwork, then $\delta\mathcal{L}\,\delta x_i = 0$. Moving left we have a $\delta\mathcal{L}$ inside the integral. The operators to the right of this can produce only linear concentration fluctuations, $\mathcal{O}(\delta x)$, because we are looking at linearised dynamics; when we apply the operator $\delta\mathcal{L}$ we then obtain zero or (if bulk terms are present) a constant. But while $\mathcal{Q}$ leaves a constant unchanged [62], the other operators $\mathcal{P}$, $\mathcal{L}_0$ and $\delta\mathcal{L}$ all annihilate it, so the leftmost $\mathcal{P}$ in (6.2.2) ensures a vanishing overall contribution. The same logic

applies to the $\delta\mathcal{L}$ in the second factor from the left. Therefore there are no changes to the structure of the memory function from the presence of the perturbation $\delta\mathcal{L}$ in the adjoint Fokker-Planck operator (6.1.3); however, compared to the entirely unregulated networks we had considered previously, there will be additional contributions in $\mathcal{QLQ}$ of $\mathcal{O}(\gamma)$ from the destruction rates. We treat these here as part of the unperturbed baseline.

Applying the same analysis to the random force we find that

$$\begin{aligned} & e^{\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\mathcal{Q}t}\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\delta x_i \\ & \quad = e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t}\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\delta x_i + \int_0^t dt'\ e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')}\mathcal{Q}(\mathcal{L}_0+\delta\mathcal{L})\delta x_i \\ & \quad = e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t}\mathcal{Q}\mathcal{L}_0\delta x_i + \int_0^t dt'\ e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')}\mathcal{Q}\mathcal{L}_0\delta x_i \end{aligned} \tag{6.2.3}$$

where the first term in the last line is the original random force and the second term is the exta contribution to the random force from the perturbation. The factor $\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')}\mathcal{Q}\mathcal{L}_0\delta x_i$ can be expressed in matrix form as follows

$$\left[\boldsymbol{\delta x}^{\mathrm{T}}\boldsymbol{Q}e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}(t-t')}\boldsymbol{Q}\boldsymbol{L}_0\right]_i \tag{6.2.4}$$

Applying $\delta\mathcal{L}$ to this gives a constant, and the next factor $\mathcal{Q}$ to the left has no effect on this. In the exponential $e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}$, every term in the power series expansion $1+\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'+\ldots$ except the zeroth order contains a factor $\mathcal{L}_0$, which will annihilate the constant. The exponential thus reduces to an identity and overall the extra contribution to the projected equations from the random force is

$$\int_0^t dt'\ \left[\boldsymbol{h}^{\mathrm{T}}e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}(t-t')}\boldsymbol{Q}\boldsymbol{L}_0\right]_i \tag{6.2.5}$$

where the contribution is $\mathcal{O}(h)$ due to the operator $\delta\mathcal{L}$.

We can then write the perturbed linear projected equations as

$$\frac{\partial}{\partial t}\delta x_i = \sum_j \delta x_j\Omega_{ji} + \sum_j\int_0^t dt'\,\delta x_j(t')M_{ji}(t-t') + \sum_k\int_0^t dt'\,h_kR_{ki}(t-t') + r_i \tag{6.2.6}$$

Here we have expressed the perturbation term (6.2.5) explicitly as

$$\boldsymbol{R}(t-t') = e^{\boldsymbol{L}^{\mathrm{b,b}}(t-t')}\boldsymbol{L}^{\mathrm{b,s}} \tag{6.2.7}$$

in terms of subblocks of the matrix representing the linearised $\mathcal{L}$. Running through the same analysis for more general time dependent perturbations (t), one finds for the perturbation term

$$\sum_k \int_0^t dt'\, h_k(t') R_{ki}(t-t') \tag{6.2.8}$$

This shows the expected time-translation invariance. It also tells us that the intuitive interpretation of R_{ki} is as an impulse response function: it represents the perturbation effect of quickly raising h_k to a nonzero value and then decreasing it again to zero. Having clarified this we go back to focussing on the case where $\boldsymbol{h}$ is constant in time, and are then led to looking at the step response function $\boldsymbol{\chi}$ that is obtained by integrating $\boldsymbol{R}$ in time:

$$\boldsymbol{\chi}(t) = \int_0^t dt'\, e^{\boldsymbol{L}^{\mathrm{b,b}}(t-t')} \boldsymbol{L}^{\mathrm{b,s}} \tag{6.2.9}$$

Overall, we see that when the creation and destruction rates balance such that $\boldsymbol{h} = 0$, there is no change to the structure of the projected equations. Though compared to an unregulated system, also the random force, like the memory functions, will be changed quantitatively by the terms in $\mathcal{L}_0$ involving the destruction rates γ_i. A nonzero $\boldsymbol{h}$, on the other hand, provides a systematic addition to the random force and this will change the steady state of the system in response to the perturbation.

In the following we will use the impulse and step response functions introduced above to analyse the subnetwork response to bulk perturbations. In so doing we will ignore the perturbation-independent piece of the random force, written as r_i above. The results then apply directly to the case where the bulk is initially in its unperturbed steady, as then $r_i(t) = 0$ at all times in linearised dynamics [62]. They also apply more generally if we think of $\boldsymbol{\delta x}$ as the difference between its value at nonzero and zero $\boldsymbol{h}$. Due to the linearity of (6.2.6), this difference obeys the same equation but with the r_i-term removed, which can be written compactly as

$$\frac{\partial}{\partial t}\boldsymbol{\delta x} = \boldsymbol{\delta x}^{\mathrm{T}} \boldsymbol{\Omega} + \int_0^t dt'\, \boldsymbol{\delta x}^{\mathrm{T}}(t') \boldsymbol{M}(t-t') + \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi}(t) \tag{6.2.10}$$

6.2.2 Perturbed steady state and time dependent response

To understand how a nonzero perturbation $\boldsymbol{h}$ causes a change in the steady state concentrations, we will assume that any species with a non-zero perturbation h_i acting on

it has a nonzero destruction rate γ_i. Otherwise the relation $h_i = \theta_i - \gamma_i y_i$ would imply that the species is created with nonzero rate θ_i but cannot be destroyed ($\gamma_i = 0$), which is a biologically unrealistic scenario.

We first sketch a naive approach to finding the new steady state of the subnetwork concentrations, without directly solving the projected equations for perturbed timecourses. One considers (6.2.6) in the limit $t \to \infty$ when $\delta x_i(t)$ becomes constant; h_i is constant by assumption. These factors can then be pulled out of the time integrals, so that at steady state the subnetwork concentrations must satisfy

$$0 = \boldsymbol{\delta x}^{\mathrm{T}} \left(\boldsymbol{\Omega} + \overline{\boldsymbol{M}} \right) + \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi}_\infty \tag{6.2.11}$$

Here $\overline{\boldsymbol{M}}$ is the memory $\boldsymbol{M}(\Delta t)$ integrated over $\Delta t = 0 \ldots \infty$, while $\boldsymbol{\chi}_\infty = \lim_{t\to\infty} \boldsymbol{\chi}(t)$ is the long time limit of the step response. Solving this equation predicts the perturbed steady state concentrations as

$$\boldsymbol{\delta x}^{\mathrm{T}} = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi}_\infty \left(-\boldsymbol{\Omega} - \overline{\boldsymbol{M}} \right)^{-1} \tag{6.2.12}$$

Here $\boldsymbol{\chi}_\infty$ gives the effect of the perturbation on the boundary: it only contains information from the matrices $\boldsymbol{L}^{\mathrm{b,b}}$ and $\boldsymbol{L}^{\mathrm{b,s}}$ and therefore represents the propagation of the perturbation in the bulk until it reaches the subnetwork species that directly interact with the bulk, i.e. the boundary. The factor $\left(-\boldsymbol{\Omega} - \overline{\boldsymbol{M}} \right)^{-1}$ then gives the effect of the resulting boundary perturbation on the subnetwork.

The analysis above is oversimplified because, in general, there will be subnetwork conservation laws that make the matrix $\left(-\boldsymbol{\Omega} - \overline{\boldsymbol{M}} \right)$ singular, hence its inverse is a priori undefined. We therefore next look at the full time course of the system transitioning from the unperturbed steady state to the perturbed one.

Let us define $\boldsymbol{U}(t)$ as the Green's function for (6.2.10). Its defining property is that, when $\boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi}(t)$ is replaced by an arbitrary perturbation $\boldsymbol{v}^{\mathrm{T}} \delta(t - t_0)$ at time t_0, the solution of the equation starting from $\boldsymbol{\delta x}(0) = 0$ is $\boldsymbol{v}^{\mathrm{T}} \boldsymbol{U}(t - t_0)$ for $t > t_0$. Then the full time dependent response to the perturbation $\boldsymbol{h}$, starting from the same initial condition, is given by the convolution

$$\boldsymbol{\delta x}^{\mathrm{T}}(t) = \int_0^t dt' \, \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi}(t') \boldsymbol{U}(t - t') \tag{6.2.13}$$

The interpretation of the two factors is as in the naive steady state analysis above: $\boldsymbol{\chi}(t)$ describes the propagation of the perturbation to the boundary, assuming the subnetwork stays at steady state, while $\boldsymbol{U}(t)$ then determines how this boundary perturbation propagates through the subnetwork.

The convolution structure of the time dependent response suggests that for further analysis one should split both $\boldsymbol{\chi}$ and $\boldsymbol{U}$ into two parts, one transient and one permanent, i.e. surviving to infinite time. We therefore rewrite (6.2.13) as

$$\boldsymbol{\delta x}^{\mathrm{T}}(t) = \int_0^t dt'\, \boldsymbol{h}^{\mathrm{T}} \left(\boldsymbol{\chi}_\infty + \Delta\boldsymbol{\chi}(t')\right) \left(\boldsymbol{U}_\infty + \Delta\boldsymbol{U}(t-t')\right) \tag{6.2.14}$$

where both $\Delta\boldsymbol{\chi}(t')$ and $\Delta\boldsymbol{U}(t-t')$ go to zero in the long time limit. Multiplying out we find that

$$\boldsymbol{\delta x}^{\mathrm{T}}(t) = \int_0^t dt'\, \boldsymbol{h}^{\mathrm{T}} \left(\boldsymbol{\chi}_\infty \boldsymbol{U}_\infty + \Delta\boldsymbol{\chi}(t')\boldsymbol{U}_\infty + \boldsymbol{\chi}_\infty \Delta\boldsymbol{U}(t-t') + \Delta\boldsymbol{\chi}(t')\Delta\boldsymbol{U}(t-t')\right) \tag{6.2.15}$$

The last contribution will vanish in the long time limit $t \to \infty$ because whatever t', at least one of the factors is always evaluated at large time argument. We will also argue more generally in Sec. 6.2.2.1 that we cannot have a contribution from the first term. Indeed the integral of this term is directly proportional to t so if it was nonzero then the perturbed system would not have a steady state. Therefore we are left with two terms that contribute to a finite steady state $\boldsymbol{\delta x}(t \to \infty)$ as shown in Figure 6.1. The top line shows the case where one only gets the combination of transient perturbation (Fig. 6.1(a)) and persistent response (Fig. 6.1(b)), while the bottom line displays the opposite combination of a persistent perturbation (Fig. 6.1(c)) together with a transient response (Fig. 6.1(d)). In general both scenarios combine to give the overall response.

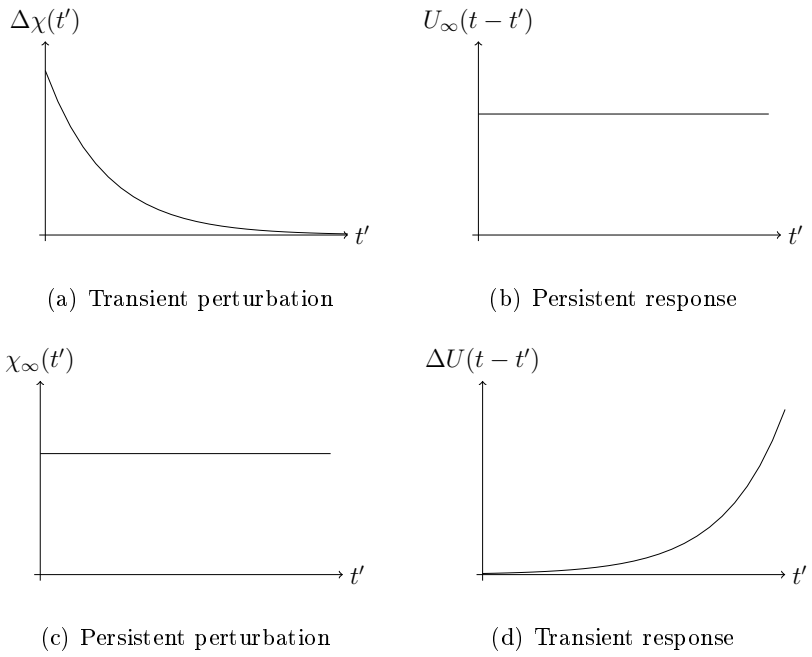
Having neglected the last term in (6.2.15) for large t, the time dependent response can be written as

$$\boldsymbol{\delta x}^{\mathrm{T}}(t) \approx \int_o^t dt'\, \boldsymbol{h}^{\mathrm{T}} \Delta\boldsymbol{\chi}(t')\boldsymbol{U}_\infty + \int_o^t dt'\, \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi}_\infty \Delta\boldsymbol{U}(t-t') \tag{6.2.16}$$

From here we can take the $t \to \infty$ limit directly to find that the perturbed steady state is given by

$$\boldsymbol{\delta x}^{\mathrm{T}} = \boldsymbol{h}^{\mathrm{T}} \left(\overline{\Delta\boldsymbol{\chi}}\boldsymbol{U}_\infty + \boldsymbol{\chi}_\infty \overline{\Delta\boldsymbol{U}}\right) \tag{6.2.17}$$

Here $\overline{\Delta\boldsymbol{\chi}}$ is the integral of $\Delta\boldsymbol{\chi}(t')$ over $t' = 0 \ldots \infty$ and similarly for $\overline{\Delta\boldsymbol{U}}$. We deduce that the naive steady state obtained by integrating the projected equations in the long time



(a) Transient perturbation (b) Persistent response

(c) Persistent perturbation (d) Transient response

Figure 6.1: The different components of $\boldsymbol{\chi}$ and $\boldsymbol{U}$. (a) Transient perturbation; (b) Persistent response; (c) Persistent perturbation; (d) Transient response. The contributions to the steady state are from (a) and (b), and (c) and (d).

limit (6.2.12) effectively assumed that the response is transient, as it only gave the term corresponding to $\boldsymbol{\chi}_\infty\overline{\Delta \boldsymbol{U}}$. Indeed, $(-\boldsymbol{\Omega}-\overline{\boldsymbol{M}})$ becomes non-invertible precisely when the response has a persistent part $\boldsymbol{U}_\infty$; this arise from conservation laws. Analysis of the full time dependent response has allowed us to obtain the additional contribution $\overline{\Delta \boldsymbol{\chi}}\boldsymbol{U}_\infty$ due to the persistent subnetwork response. We will see in Section 6.2.4 that this "additional" contribution $\overline{\Delta \boldsymbol{\chi}}\boldsymbol{U}_\infty$ is in fact present quite generically, while the contribution $\boldsymbol{\chi}_\infty\overline{\Delta \boldsymbol{U}}$ only appears under certain conditions.

6.2.2.1 Laplace transform analysis

Before applying the insights developed above, we show in this section explicitly that the first term in (6.2.15) does indeed vanish, which is what lead us to the separation of the late time response into two contributions as in (6.2.16), and the corresponding result (6.2.17) for the perturbed steady state.

Let $\hat{\boldsymbol{\chi}}(z)$ be the Laplace transform of $\boldsymbol{\chi}(t)$ and similarly $\hat{\boldsymbol{U}}(z)$ the Laplace transform of $\boldsymbol{U}(t)$. Rewriting equation (6.2.13) in Laplace space then gives the Laplace transform of

the time time dependent response $\boldsymbol{\delta x}(t)$ as

$$\boldsymbol{\delta}\hat{\boldsymbol{x}}(z) = \boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{\chi}}(z)\hat{\boldsymbol{U}}(z) \tag{6.2.18}$$

Now $\boldsymbol{U}(t)$ was defined as the Green's function of the unperturbed projected equations, involving the rate matrix and memory function. Therefore we can write the Laplace Transform of $\boldsymbol{U}(t)$ as

$$\hat{\boldsymbol{U}}(z) = \Big(z - \boldsymbol{\Omega} - \hat{\boldsymbol{M}}(z)\Big)^{-1} \tag{6.2.19}$$

Inserting then explicitly the Laplace transform of the memory function from (5.2.6) gives

$$\begin{aligned}
\hat{\boldsymbol{U}}(z) &= \Big(z - \boldsymbol{L}^{\mathrm{s,s}} - \boldsymbol{L}^{\mathrm{s,b}}(z - \boldsymbol{L}^{\mathrm{b,b}})^{-1}\boldsymbol{L}^{\mathrm{b,s}}\Big)^{-1} \\
&= \begin{pmatrix} z - \boldsymbol{L}^{\mathrm{s,s}} & -\boldsymbol{L}^{\mathrm{s,b}} \\ -\boldsymbol{L}^{\mathrm{b,s}} & z - \boldsymbol{L}^{\mathrm{b,b}} \end{pmatrix}^{-1}_{\mathrm{ss}} \\
&= \big((z - \boldsymbol{L})^{-1}\big)^{\mathrm{ss}}
\end{aligned} \tag{6.2.20}$$

Intuitively, then, $\hat{\boldsymbol{U}}(z)$ just gives the subnetwork block of the full network dynamics. Similarly taking the Laplace Transform of $\boldsymbol{R}(t)$ and $\boldsymbol{\chi}(t)$ from (6.2.7) and (6.2.9) we find

$$\begin{aligned}
\hat{\boldsymbol{R}}(z) &= (z - \boldsymbol{L}^{\mathrm{b,b}})^{-1}\boldsymbol{L}^{\mathrm{b,s}} \\
\hat{\boldsymbol{\chi}}(z) &= \frac{1}{z}\hat{\boldsymbol{R}}(z)
\end{aligned} \tag{6.2.21}$$

We are interested in the long time limit, corresponding to $z \to 0$. In $\hat{\boldsymbol{U}}(z)$, the most divergent contribution will be of order $1/z$, arising from zero eigenvalues of $\boldsymbol{L}$. We therefore write

$$\hat{\boldsymbol{U}} = \frac{1}{z}\boldsymbol{U}_\infty + \Delta\hat{\boldsymbol{U}}(z) \tag{6.2.22}$$

where $\Delta\hat{\boldsymbol{U}}(z)$ stays finite for $z \to 0$. An analogous decomposition can be made for $\hat{\boldsymbol{R}}(z)$. As $\hat{\boldsymbol{\chi}}(z) = z^{-1}\hat{\boldsymbol{R}}(z)$, its most divergent term is potentially of order $1/z^2$, so we expand

$$\hat{\boldsymbol{\chi}} = \frac{1}{z^2}\hat{\boldsymbol{\chi}}_{\mathrm{lin}} + \frac{1}{z}\boldsymbol{\chi}_\infty + \Delta\hat{\boldsymbol{\chi}}(z) \tag{6.2.23}$$

Here $\hat{\boldsymbol{\chi}}_{\mathrm{lin}}$ corresponds to a contribution to $\boldsymbol{\chi}(t)$ increasing linearly in t, but we will argue below that $\boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{\chi}}_{\mathrm{lin}} = 0$ so that this term can be removed. Once this is done, $\boldsymbol{\chi}_\infty$ and $\boldsymbol{U}_\infty$ are the long time limits of $\boldsymbol{\chi}(t)$ and $\boldsymbol{U}(t)$ respectively while $\Delta\hat{\boldsymbol{\chi}}$ and $\Delta\hat{\boldsymbol{U}}$ are the (Laplace transforms of the) transient parts.

The long-time limit $\boldsymbol{\delta x}(t \to \infty)$ is obtained by taking $\lim_{z\to 0} z\boldsymbol{\delta \hat{x}}(z)$. Inserting the above decompositions of $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{\chi}}$ into (6.2.18), we can express the perturbed steady state as

$$\lim_{z\to 0} z\boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{\chi}}\hat{\boldsymbol{U}} = \lim_{z\to 0}\Big(\frac{1}{z}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\chi}_\infty\boldsymbol{U}_\infty + \boldsymbol{h}^{\mathrm{T}}\Delta\hat{\boldsymbol{\chi}}(z)\boldsymbol{U}_\infty + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{\chi}_\infty\Delta\hat{\boldsymbol{U}}(z) + \mathcal{O}(z)\Big) \tag{6.2.24}$$

We show below that the first term vanishes. This then leaves in the brackets precisely the Laplace-domain version of the two-contribution split (6.2.16) that we wanted to demonstrate. The corresponding decomposition for the perturbed steady state (6.2.17) follows directly for $z \to 0$ because $\Delta\hat{\boldsymbol{U}}(z \to 0) = \overline{\Delta\boldsymbol{U}}$ and similarly $\Delta\hat{\boldsymbol{\chi}}(z \to 0) = \overline{\Delta\boldsymbol{\chi}}$.

To complete the arguments above, we need to show that $\boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{\chi}}_{\mathrm{lin}}$ and $\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\chi}_\infty\boldsymbol{U}_\infty$ are both zero. We will need the assumption set out above, namely that any bulk species i that is being perturbed ($h_i \neq 0$) also has a nonzero destruction rate γ_i. Our reasoning rests on the fact that such species can then not form part of any conserved densities.

We will need to be able to write down the divergent (for $z \to 0$) and finite contributions to $\hat{\boldsymbol{\chi}}$ and $\hat{\boldsymbol{U}}$. Starting with the former, let $\bar{\boldsymbol{r}}$ and $\bar{\boldsymbol{l}}$ be the right and left eigenvectors and μ be the eigenvalues of the matrix $\boldsymbol{L}^{\mathrm{b,b}}$. Indexing both by β, we write the Laplace transform (6.2.21) of the impulse response function as

$$\begin{aligned}\hat{\boldsymbol{R}}(z) &= \frac{1}{z}\sum_{\beta,\mu_\beta=0}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} + \sum_{\beta,\mu_\beta\neq 0}\frac{1}{z-\mu_\beta}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} \\ &= \frac{1}{z}\sum_{\beta,\mu_\beta=0}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} - \sum_{\beta,\mu_\beta\neq 0}\frac{1}{\mu_\beta}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} - \sum_{\beta,\mu_\beta\neq 0}\frac{z}{\mu_\beta(\mu_\beta - z)}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}}\end{aligned} \tag{6.2.25}$$

We can then identify the terms in the decomposition (6.2.23) of the Laplace transformed step response as

$$\hat{\boldsymbol{\chi}}_{\mathrm{lin}} = \sum_{\beta,\mu_\beta=0}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} \tag{6.2.26}$$

$$\boldsymbol{\chi}_\infty = -\sum_{\beta,\mu_\beta\neq 0}\frac{1}{\mu_\beta}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} \tag{6.2.27}$$

$$\Delta\hat{\boldsymbol{\chi}} = -\sum_{\beta,\mu_\beta\neq 0}\frac{1}{\mu_\beta(\mu_\beta - z)}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}_\beta^{\mathrm{T}}\boldsymbol{L}^{\mathrm{b,s}} \tag{6.2.28}$$

To argue our first desired result, $\boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{\chi}}_{\mathrm{lin}} = 0$ it then suffices to show that $\boldsymbol{h}^{\mathrm{T}}\bar{\boldsymbol{r}}_\beta = 0$ whenever $\mu_\beta = 0$. This is true because $\boldsymbol{L}^{\mathrm{b,b}}$ describes the linearised dynamics of the bulk when the subnetwork variables are clamped to $\boldsymbol{\delta x} = 0$. The $\bar{\boldsymbol{r}}_\beta$ with $\mu_\beta = 0$ define

conserved densities $\boldsymbol{\delta x}^{\mathrm{b}\mathrm{T}}\bar{\boldsymbol{r}}_\beta$ under this dynamics. But perturbed species cannot feature in conservation laws, so their coefficients in such eigenvectors $\bar{\boldsymbol{r}}_\beta$ must vanish, hence $\boldsymbol{h}^{\mathrm{T}}\bar{\boldsymbol{r}}_\beta = 0$.

For our second claim $\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\chi}_\infty\boldsymbol{U}_\infty = 0$ we need a similar decomposition of $\hat{\boldsymbol{U}}$ as given in (6.2.20). Let then $\boldsymbol{r}^{\mathrm{s}}$ and $\boldsymbol{l}^{\mathrm{s}}$ be the subnetwork components of the right and left eigenvectors of $\boldsymbol{L}$ and λ the corresponding eigenvalues, indexed by α. We can write $\hat{\boldsymbol{U}}$ as

$$\hat{\boldsymbol{U}} = \sum_{\alpha,\lambda_\alpha=0} \frac{1}{z}\boldsymbol{r}^{\mathrm{s}}_\alpha \boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha + \sum_{\alpha,\lambda_\alpha\neq 0} \frac{1}{z-\lambda_\alpha}\boldsymbol{r}^{\mathrm{s}}_\alpha \boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha \tag{6.2.29}$$

The contributions in (6.2.22) are then

$$\boldsymbol{U}_\infty = \sum_{\alpha,\lambda_\alpha=0} \boldsymbol{r}^{\mathrm{s}}_\alpha \boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha \tag{6.2.30}$$

$$\Delta\hat{\boldsymbol{U}} = -\sum_{\alpha,\lambda_\alpha\neq 0} \frac{1}{\lambda_\alpha - z}\boldsymbol{r}^{\mathrm{s}}_\alpha \boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha \tag{6.2.31}$$

Using (6.2.27) and (6.2.30) we find

$$\begin{aligned}
\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\chi}_\infty\boldsymbol{U}_\infty &= -\sum_{\beta,\mu_\beta\neq 0}\sum_{\alpha,\lambda_\alpha=0} \boldsymbol{h}^{\mathrm{T}}\frac{1}{\mu_\beta}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta\boldsymbol{L}^{\mathrm{b,s}}\boldsymbol{r}^{\mathrm{s}}_\alpha\boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha \\
&= \sum_{\beta,\mu_\beta\neq 0}\sum_{\alpha,\lambda_\alpha=0} \boldsymbol{h}^{\mathrm{T}}\frac{1}{\mu_\beta}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta\boldsymbol{L}^{\mathrm{b,b}}\boldsymbol{r}^{\mathrm{b}}_\alpha\boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha \\
&= \sum_{\beta,\mu_\beta\neq 0}\sum_{\alpha,\lambda_\alpha=0} \boldsymbol{h}^{\mathrm{T}}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta\boldsymbol{r}^{\mathrm{b}}_\alpha\boldsymbol{l}^{\mathrm{s}\,\mathrm{T}}_\alpha \\
&= \boldsymbol{h}^{\mathrm{T}}\left(\sum_{\beta,\mu_\beta\neq 0}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta + \sum_{\beta,\mu_\beta=0}\bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta\right)\left(\sum_{\alpha,\lambda_\alpha=0}\boldsymbol{r}^{\mathrm{b}}_\alpha\boldsymbol{l}^{\mathrm{s}^{\mathrm{T}}}_\alpha\right) \\
&= \boldsymbol{h}^{\mathrm{T}}\sum_{\alpha,\lambda_\alpha=0}\boldsymbol{r}^{\mathrm{b}}_\alpha\boldsymbol{l}^{\mathrm{s}^{\mathrm{T}}}_\alpha \\
&= 0
\end{aligned}$$

which is our desired result. We have first used the fact that $\boldsymbol{L}^{\mathrm{b,s}}\boldsymbol{r}^{\mathrm{s}}_\alpha + \boldsymbol{L}^{\mathrm{b,b}}\boldsymbol{r}^{\mathrm{b}}_\alpha = 0$ for any right eigenvector of $\boldsymbol{L}$ with eigenvalue $\lambda_\alpha = 0$. This identity allows one to eliminate $\boldsymbol{L}^{\mathrm{b,s}}$ in favour of $\boldsymbol{L}^{\mathrm{b,b}}$, which can then be used to simplify $\mu_\beta^{-1}\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta\boldsymbol{L}^{\mathrm{b,b}} = \bar{\boldsymbol{l}}^{\mathrm{T}}_\beta$. In the next step we added terms involving the $\boldsymbol{L}^{\mathrm{b,b}}$-eigenvectors with vanishing eigenvalue μ_β, which vanish because $\boldsymbol{h}^{\mathrm{T}}\bar{\boldsymbol{r}}_\beta = 0$ as argued above. With the sum over β including all eigenvectors, $\sum_\beta \bar{\boldsymbol{r}}_\beta\bar{\boldsymbol{l}}^{\mathrm{T}}_\beta$ is then the identity matrix. Finally $\boldsymbol{h}^{\mathrm{T}}\boldsymbol{r}^{\mathrm{b}}_\alpha = 0$ for $\boldsymbol{L}$-eigenvectors

with $\lambda_\alpha = 0$, because again such eigenvectors contain coefficients defining conserved species – this time of the full dynamics defined by $\boldsymbol{L}$ – and perturbed species cannot appear in conservation laws.

6.2.3 Single versus multiple perturbed species

The result (6.2.17) decomposing the perturbed steady state into two separate contributions works generally, whether we are dealing with perturbations from gene regulation to a single or multiple species. There is an additional consideration, however, in that we would like to be able to relate the size of the perturbation(s) h_i to the changes of the concentration of bulk species i, as that is a convenient way of characterizing the strength of a perturbation in e.g. knockdown experiments.

To make this task simple we will focus primarily on systems where only one species, say i, is regulated while all others have $h_j = \gamma_j = 0$. The perturbed steady state density of the regulated species is then just

$$\delta x_i = \frac{h_i}{\gamma_i} \tag{6.2.32}$$

To see why this is so, consider the unregulated system, i.e. (6.1.1) without $h_i - \gamma_i \delta x_i$. We assume that species i features in at least one conserved density X in this unregulated system; this is not something we can prove in general but it should hold in any biologically reasonable system of unary and binary reactions. In the unregulated system, then, $\partial X/\partial t = 0$. The regulated (and perturbed) system only differs by the addition of the term $h_i - \gamma_i \delta x_i$ in the equation of motion for δx_i. This changes the evolution equation for X to $\partial X/\partial t = c(h_i - \gamma_i \delta x_i)$ where c is the stochiometric coefficient with which δx_i features in the conserved density X (e.g. $c = 1$ in the simplest case). But in the perturbed steady state we have to have $\partial X/\partial t = 0$ again, and so (6.2.32) follows.

One sees from the above also why the case of multiple perturbed species is more complicated: here $\partial X/\partial t$ acquires several additional contributions in the regulated system, which in the new steady state do not have to – and in general will not – vanish individually. A case where this is still true is the one where one has at least as many conservation laws in the unregulated systems as perturbed species, and the coefficient vectors with which these species enter in the conservation laws are linearly independent. But this is a condition that would have to be checked explicitly for any given reaction network.

6.2.4 Examples

We have seen that there are two potential contributions to the perturbed steady state contributions but now want to get further insight into when these contributions appear in practice and what their relative importance is. To do this we will look at some example networks and analyse the different contributions obtained from $\boldsymbol{\chi}$ and $\boldsymbol{U}$. In the graphical representations we are generally using dimensionless quantities and therefore are plotting $\tilde{\chi}_{ki}(t) = y_k \chi_{ki}(t) y_i^{-1}$ and $\tilde{U}_{ij} = y_i U_{ij} y_j^{-1}$.

We first consider the example network shown in Fig. 6.2(a), with three complex formation and dissociation reactions as indicated below:

$$\begin{aligned} 1 + 2 &\rightleftharpoons 3 \\ 1 + 4 &\rightleftharpoons 5 \\ 2 + 4 &\rightleftharpoons 6 \end{aligned} \tag{6.2.33}$$

The subnetwork contains species 1, 2 and 3 and the bulk contains species 4, 5 and 6.

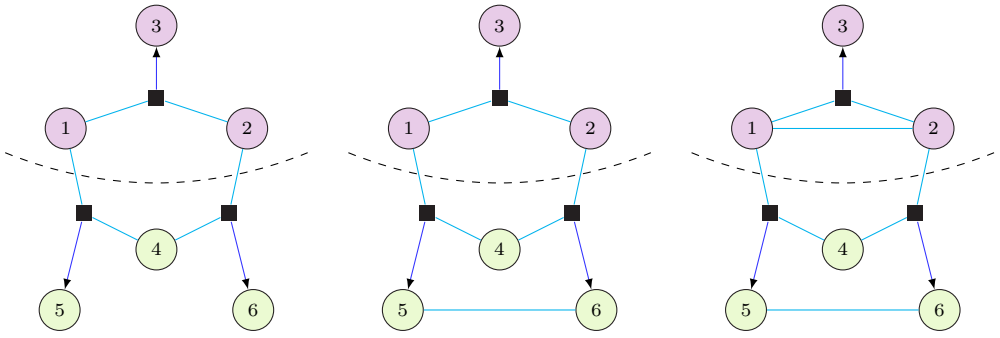


Figure 6.2: Sketch of three simple model protein interaction networks. (a) Protein 1 reacts with protein 2 to form complex 3, and in reverse 3 can dissociate into 1 and 2. There are two analogous reactions where 1 reacts with 4 to form 5 and 2 reacts with 4 to form 6, and the reverse dissociations. The subnetwork species were chosen to be species 1, 2 and 3 and the bulk to be 4, 5 and 6; (b) An extension of (a) with a conformational change between species 5 and 6; (c) An extension of (b) with an additional conformational change between species 1 and 2.

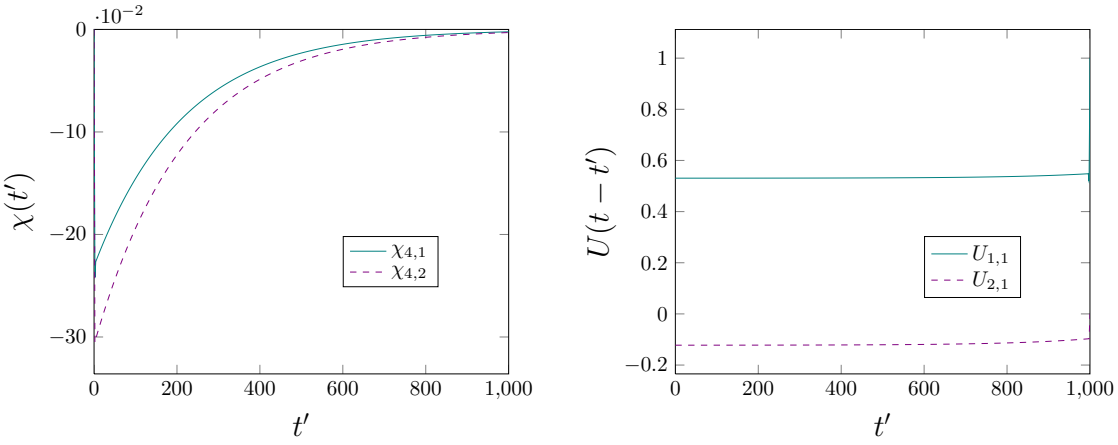
For this schematic example we work in dimensionless units for times and concentrations.

When the subnetwork is at steady state there is detailed balance in the bulk (excluding

the perturbation term). There is also detailed balance in the subnetwork so that for any possible steady state all the reactions inside the subnetwork have zero flux. We can see this by looking at the separate reactions. Species 3 in the subnetwork changes only because of the flux of the reaction between 1 and 2 (more precisely the net flux, i.e. the difference between the forward and backward reaction fluxes), so this flux must be individually zero. The same is true for the reaction fluxes affecting species 5 and 6 in the bulk. We perturb species 4 and look at how this affects the subnetwork. Throughout this section we will look at the contributions from $\boldsymbol{\chi}$ and $\boldsymbol{U}$ that govern the perturbed steady state of species 1. Fig. 6.3 shows the contributions from $\boldsymbol{\chi}(t')$ and $\boldsymbol{U}(t-t')$ that affect the steady state of protein 1, where $\boldsymbol{\chi}$ represents the change in the bulk dynamics with the subnetwork clamped to steady state and $\boldsymbol{U}$ tells us how this change then propagates from the boundary through the subnetwork. We can see that the only contribution to the steady state will be from $\overline{\boldsymbol{\chi}}\boldsymbol{U}_\infty$. When species 4 is perturbed, because of the detailed balance in the bulk, the bulk reaches a steady state without any boundary fluxes, where a boundary flux is a nonzero (net) reaction flux in a reaction across the boundary and is represented by a nonzero $\boldsymbol{\chi}_\infty$ contribution. Therefore $\boldsymbol{\chi}(t')$ approaches zero and we only obtain contributions from the transient part of $\boldsymbol{\chi}(t')$ and the persistent part of $\boldsymbol{U}(t-t')$.

We next extend this example to include a conformational change between species 5 and 6 in the bulk as shown in Fig. 6.2(b). There is now no detailed balance in the bulk when the subnetwork is clamped to steady state; however Fig. 6.4 shows that we still only obtain a contribution to the perturbed steady state from the transient part of $\boldsymbol{\chi}$ and the persistent part of $\boldsymbol{U}$. Without detailed balance in the bulk we might have expected a nonzero $\boldsymbol{\chi}_\infty$ term because we can now have a nonzero boundary flux. However, although nonzero boundary fluxes are possible in principle, if we had a flux across the boundary it would lead to a continual increase in δx_1 or δx_2. As these species interact only via the formation of species 3, the subnetwork could not then reach a steady state. Therefore there can be no boundary fluxes in the perturbed steady state and we again obtain only the contribution from $\overline{\boldsymbol{\chi}}\boldsymbol{U}_\infty$.

The third example we consider extends the model further to include a conformational change in the subnetwork between species 1 and 2 as shown in Fig. 6.2(c). There is no

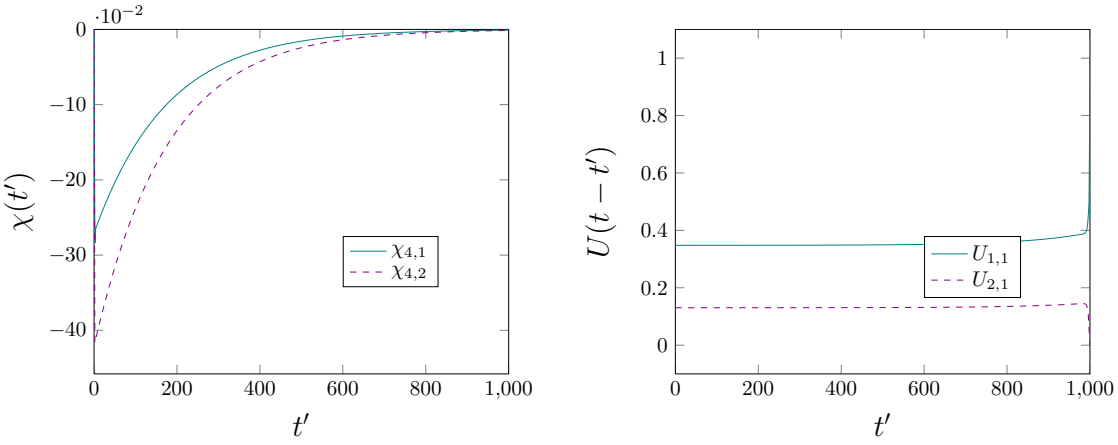


(a) (b)

Figure 6.3: Plots of perturbation and response for the example in Fig. 6.2(a). (a) The contributions from $\boldsymbol{\chi}$ on the boundary species when 4 is perturbed. (b) The contributions from $\boldsymbol{U}$ that are needed to obtain the steady state of species 1. $\delta x_1(t) = \int_0^t dt' \ (h_4 \chi_{4,1}(t') U_{1,1}(t-t') + h_4 \chi_{4,2}(t') U_{2,1}(t-t'))$. There are no boundary fluxes and hence all components of $\chi(t')$ decay to zero; as a consequence the perturbed steady state of species 1 has a contribution only from $\overline{\boldsymbol{\chi}}\boldsymbol{U}_\infty$. Parameters: $t = 1000$, reaction rates $k^+_{12,3} = k^+_{14,5} = 1$, $k^-_{3,12} = k^-_{5,14} = k^+_{24,6} = 2$, $k^-_{6,24} = 3$, $h = 0.001$ and $\gamma = 0.01$.

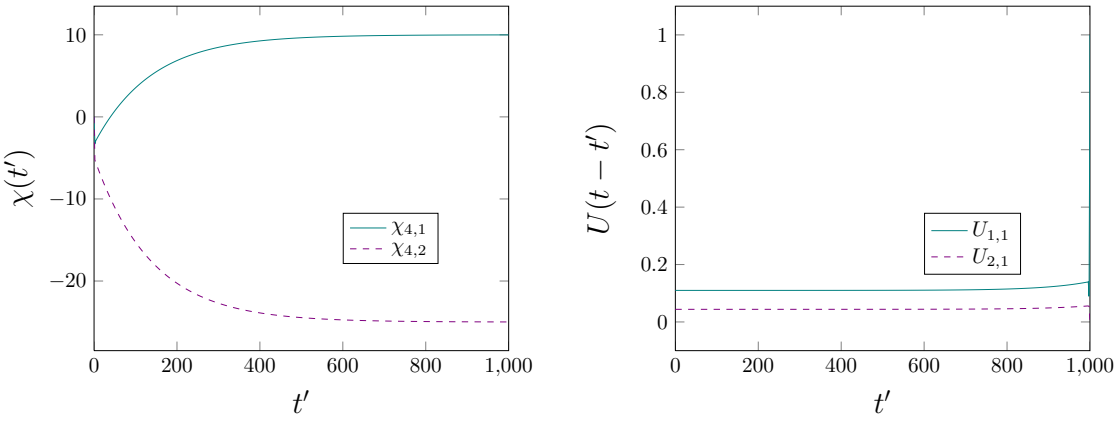
detailed balance in the bulk and for generic rate parameters there is also a flux through the subnetwork. Fig. 6.5 shows that now $\boldsymbol{\chi}(t')$ has a persistent part and therefore we obtain contributions to the steady state from both $\overline{\boldsymbol{\chi}}\boldsymbol{U}_\infty$ and $\boldsymbol{\chi}_\infty \overline{\boldsymbol{U}}$. Overall, we see that if a system has detailed balance in the bulk then we will only get transient boundary perturbations, but in order to obtain persistent perturbations we need a "flux loop" across the boundary. Here we use the term "flux loop" to indicate any network of boundary fluxes that travel both in and out of the subnetwork.

Whether a boundary species has a nonzero reaction flux can be more clearly seen by looking directly at the fluxes of each reaction across the boundary. For a system with no "flux loops" across the boundary we would expect that the total flux across the boundary would be zero both before and after perturbation. However for a system with both a persistent and transient contribution from $\boldsymbol{\chi}$ we would expect to see a positive flux through part of the boundary and a negative flux through other boundary species, both before and after perturbation, indicating a "flux loop".



(a) (b)

Figure 6.4: Plots of perturbation and response for the example in Fig. 6.2(b). (a) The contributions from $\boldsymbol{\chi}$ on the boundary species when 4 is perturbed. (b) The contributions from $\boldsymbol{U}$ that are needed to obtain the steady state of species 1. There are no "flux loops" and therefore there is only a contribution from $\overline{\boldsymbol{\chi}}\boldsymbol{U}_\infty$. Reaction rates and other parameters as in Fig. 6.3 with $\lambda_{56} = 0.5, \lambda_{65} = 1$.



(a) (b)

Figure 6.5: Plots of perturbation and response for the example in Fig. 6.2(c). (a) The contributions from $\boldsymbol{\chi}$ on the boundary species when 4 is perturbed are persistent. (b) The contributions from $\boldsymbol{U}$ that are needed to obtain the steady state of species 1. There is a contribution from both $\overline{\boldsymbol{\chi}}\boldsymbol{U}_\infty$ and $\boldsymbol{\chi}_\infty\overline{\boldsymbol{U}}$. Reaction rates and other parameters as in Fig. 6.4 with $\lambda_{12} = 0.5, \lambda_{21} = 1$.

For the systems in Fig. 6.2 the flux across the boundary into species 1 is $k^+_{14,5}x_1x_4 - k^-_{5,14}x_5$ and similarly the flux into species 2 is $k^+_{24,6}x_2x_4 - k^-_{6,24}x_6$. The systems in

Figs. 6.2(a) and 6.2(b) both have zero fluxes into species 1 and 2 both before and after a perturbation. However, it can be seen in Fig. 6.6 that the example in Fig. 6.2(c) has a negative flux into species 1 and a positive flux out of species 2 both before and after perturbation creating a "flux loop" across the boundary.

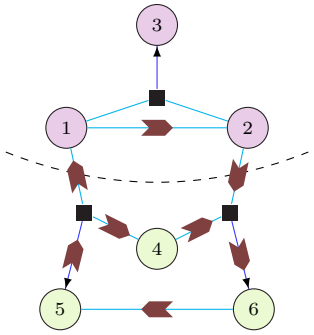


Figure 6.6: The "flux loop" of the example from Fig. 6.2(c). The arrows indicate the direction of the fluxes. Before perturbation the flux is 0.05. The flux changes linearly with h when the system is perturbed; quantitatively (flux before - flux after perturbation)$/h = 0.307$.

6.2.5 Application to EGFR

We have shown that the response of the subnetwork to a bulk perturbation can be split into two contributions. We now consider applying our results to the model of epidermal growth factor receptor detailed in Section 4. Initially we focus on perturbing the bulk species Shc. We choose a destruction rate $\gamma_{\text{Shc}} = 0.01\text{s}^{-1}$ and, where a concrete perturbation size is needed, $h_{\text{Shc}}/y_{\text{Shc}} = 0.001\text{s}^{-1}$ so that from (6.2.32) the perturbation increases the steady state concentration of Shc by 10%. The chosen regulation timescale is $1/\gamma_{\text{Shc}} = 100\text{s}$, which is longer than for the typical protein reactions in the EGFR network model [13] but still short in absolute terms. Smaller values of γ_{Shc} give qualitatively similar results; however the dynamics take place on the regulation timescale. Therefore decreasing the size of γ_{Shc} and h_{Shc} while keeping the size of the perturbation fixed will result in a longer time taken to reach the steady state.

Looking at the contributions to the new steady state from $\boldsymbol{\chi}$ and $\boldsymbol{U}$ we notice that within the network we obtain two different types of contributions from the perturbation. The boundary perturbation $\boldsymbol{\chi}$ from Shc on phosphorylated EGFR (RP) is purely transient

whereas the contributions from $\boldsymbol{\chi}$ which give the effect of the perturbation onto the other boundary species also have a persistent part. This can be seen in Fig. 6.7. The perturbed

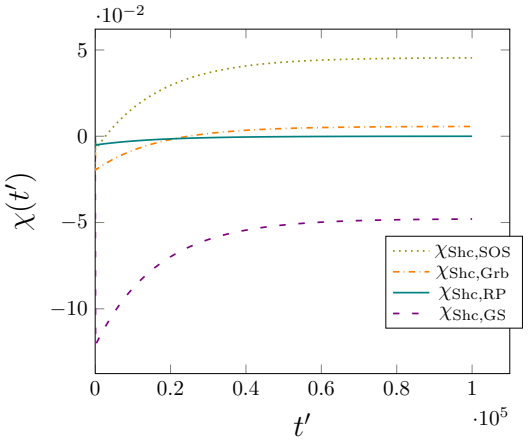
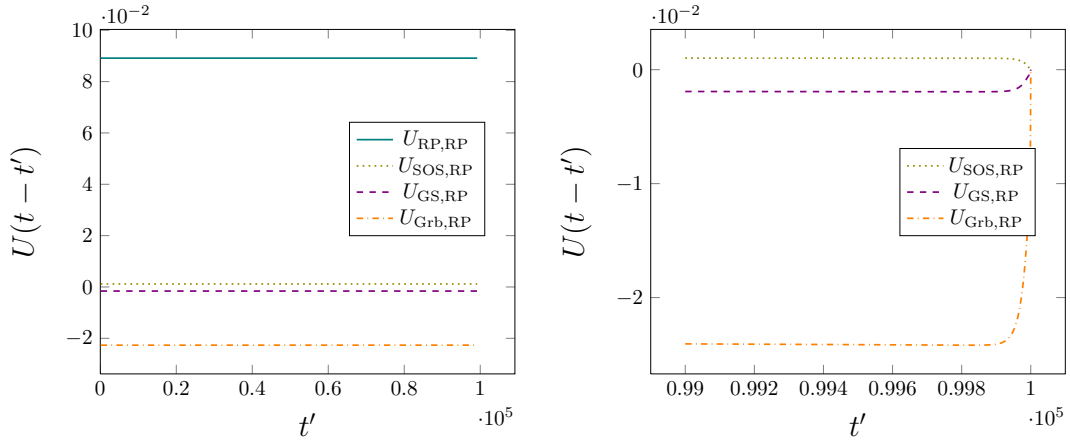


Figure 6.7: The boundary perturbation $\chi_{\mathrm{Shc,RP}}$ is purely transient, while the perturbation of the other boundary species, Grb2, SOS and Grb2-SOS (GS), also have a persistent part.

steady state of the subnetwork species will have a contribution from $\Delta\bar{\chi}_{\mathrm{Shc,RP}}U_{\infty_{\mathrm{RP,s}}}$, where s denotes the subnetwork species of interest. This contribution will be from the persistent part of $U_{\mathrm{RP,s}}$ but not the transient part, because there is no persistent contribution from $\chi_{\infty_{\mathrm{Shc,RP}}}$. The contributions to the steady state through the other boundary species will feature both transient and persistent parts from both the boundary perturbation and subnetwork response. Fig. 6.8 shows the contributions from $\boldsymbol{U}$ that are needed to obtain the steady state of phosphorylated EGFR (RP).

The reason that Grb2, SOS and Grb2-SOS (GS) have a persistent contribution to the perturbation is that there is a flux loop through these reactions across the boundary between subnetwork and bulk. If we look at the fluxes across the boundary before perturbation as shown in Fig. 6.9 we find that the flux through phosphorylated EGFR (RP) is zero, but there is a small flux through the other boundary species. Looking at the subnetwork fluxes into Grb2, SOS and out of Grb2-SOS (GS) we see that for the system to be in steady state there must be a boundary flux into Grb2 and SOS and out of Grb2-SOS (GS). This leads to a flux loop through the three species, and the boundary perturbations of these species can therefore have persistent parts. On the other hand one has only a transient contribution to the boundary perturbation on phosphorylated



(a) (b)

Figure 6.8: The contributions from $\boldsymbol{U}$ that are needed to obtain the steady state of phosphorylated EGFR (RP): (a) persistent contributions; (b) zoom on the region t' close to t, to make the transient contributions visible. $t = 10^5$s, $\gamma = 0.01\text{s}^{-1}$ and $h/y = 0.001\text{s}^{-1}$.

EGFR (RP).

6.2.5.1 Changing the perturbed species

We can use plots of the matrices $\Delta\boldsymbol{\chi}$ and $\boldsymbol{U}_\infty$, and $\boldsymbol{\chi}_\infty$ and $\Delta\boldsymbol{U}$ as a simple way to obtain an understanding of how the perturbation propagates to the boundary and then from there affects the subnetwork. We focus in the following on $\Delta\boldsymbol{\chi}$ and $\boldsymbol{U}_\infty$ because for the EGFR network the combination of these two terms gives the dominant contribution to the overall steady state response to the perturbation.

So far we have looked at perturbing Shc. Among the four boundary species of our subnetwork, this only interacts directly with phosphorylated EGFR (RP), which receives no net flux from the bulk. Therefore it is harder to interpret the effects of Shc as it only has an indirect effect on most of the boundary species. We therefore consider perturbing two other species which interact more strongly with the subnetwork: Shc-Grb2 (ShG), which interacts directly with phosphorylated EGFR (RP), Grb2 and SOS, as well as phosphorylated Shc (ShP), which interacts directly with phosphorylated EGFR (RP), Grb2 and Grb2-SOS (GS). Phosphorylated Shc (ShP) reacts with Grb2 to make Shc-

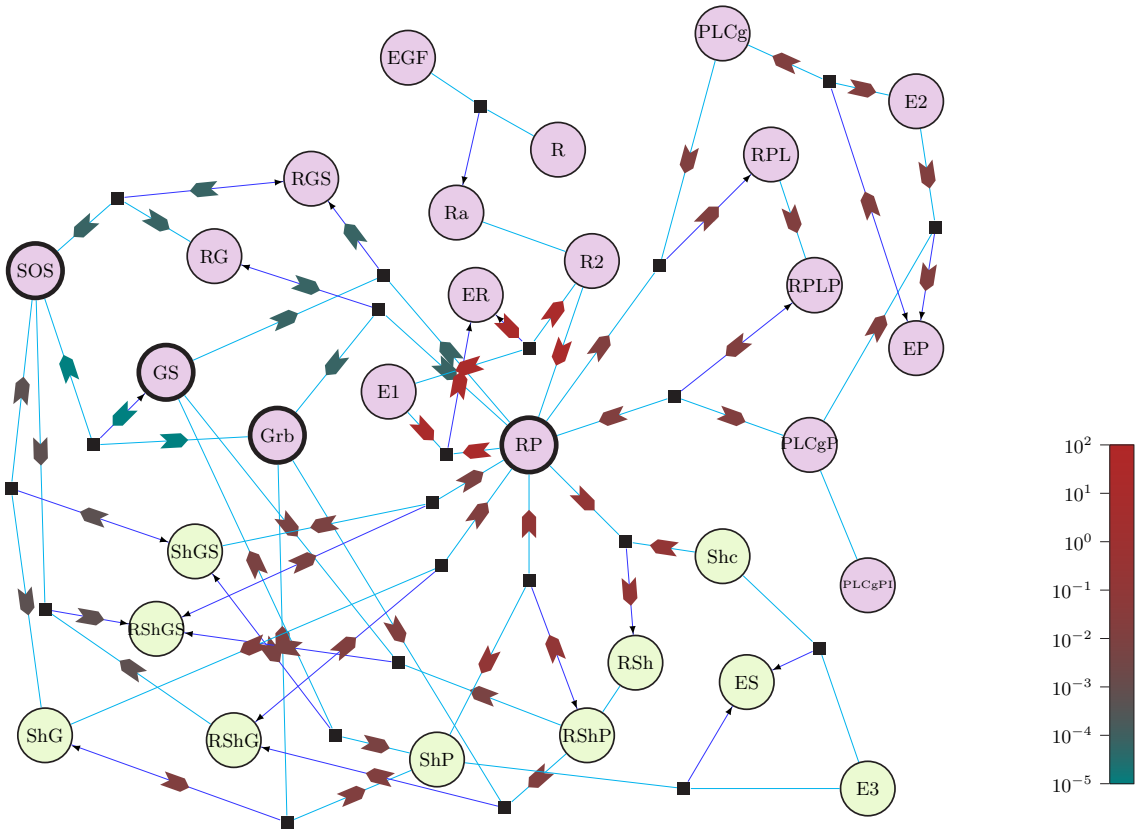


Figure 6.9: Net fluxes across the subnetwork and boundary of the EGFR network. The arrows represent the direction of the fluxes and the colour scale represents the flux strength. The fluxes through phosphorylated EGFR (RP) balance, but there is a loop of (weak) net fluxes through Grb2, SOS and Grb2-SOS (GS).

Grb2 (ShG); thus expect the entries of $\boldsymbol{\chi}$ for the perturbation effect on Grb2 to be rather different in the case where we perturb ShP (a reactant) or ShG (a product). Figure 6.10(a) shows the vectors $\boldsymbol{h}^{\mathrm{T}}\Delta\boldsymbol{\chi}$ when the perturbed species is Shc, Shc-Grb2 (ShG) and phosphorylated Shc (ShP). All systems have $\gamma = 0.01\mathrm{s}^{-1}$ and $h/y = 0.005\mathrm{s}^{-1}$. The results are intuitively reasonable: a perturbation towards increasing concentration of ShG has a positive effect on Grb2 (more ShG can decompose into Grb2), whereas a perturbation creating more ShP has a negative effect on Grb2 (more ShP to soak up Grb2 and produce ShG). We can also see that after a perturbation of Shc there is a negative effect on phosphorylated EGFR (RP) as more Shc will react with RP to make RSh; however there is also a negative effect on the other boundary species which is harder to interpret in this way. We see in Figures 6.10(b) and 6.10(c) that the matrices $\boldsymbol{U}_{\infty}$ encode similar effects for propagation of the perturbation from the boundary into

the subnetwork, though with some quantitative differences. The latter arise from the fact that the unperturbed system, whose behaviour $\boldsymbol{U}_\infty$ describes, is slightly different in the two cases, one having a nonzero destruction rate for ShG and the other for ShP. The plots of $\boldsymbol{U}_\infty$ have some noticeable features such as the "self-response" terms, i.e. the effect of a boundary species on itself is always positive. Similarly we can see that a boundary perturbation of Grb2 has a (weak) negative effect on SOS but a (weak) positive effect on Grb2-SOS (GS).This is to be expected because Grb2 reacts with SOS to make GS, hence we would predict that an increase in Grb2 would decrease the amount of SOS and increase the amount of GS.

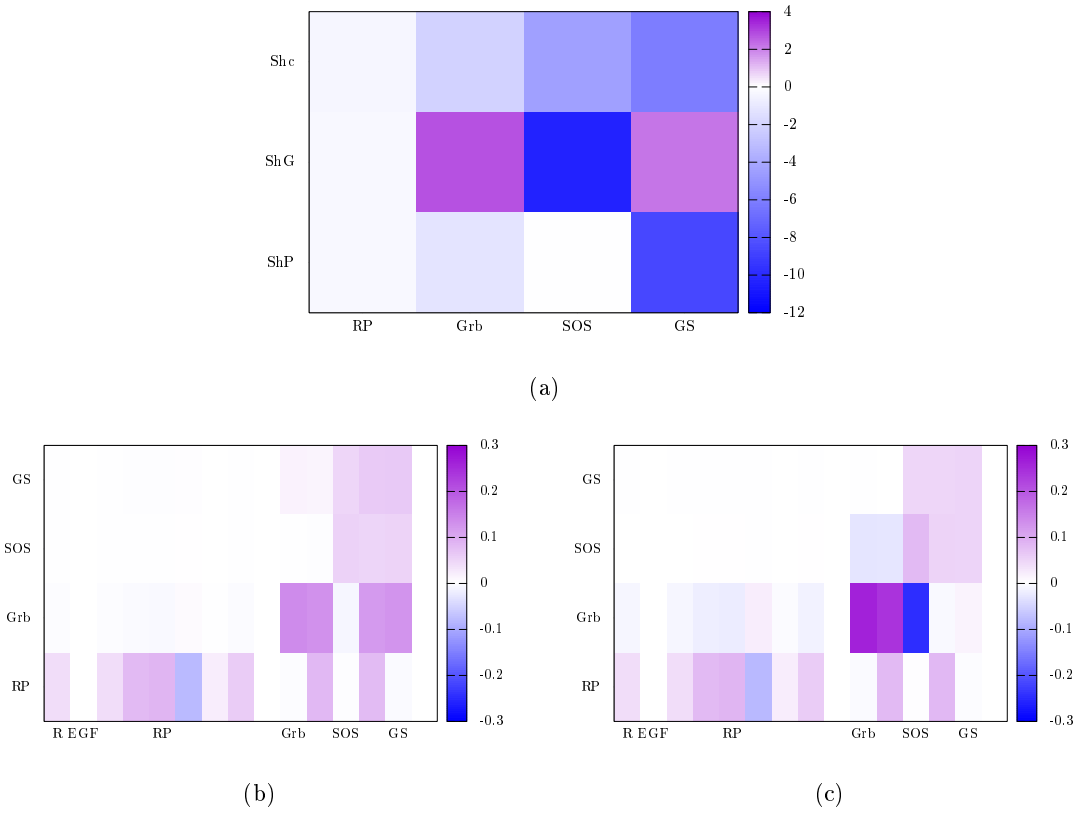


Figure 6.10: Matrix plots of (a)$\boldsymbol{h}^{\mathrm{T}}\Delta\boldsymbol{\chi}$ from systems with Shc, ShG and ShP as the perturbed species; (b)$\boldsymbol{U}_\infty$ when ShG is the perturbed species; (c) $\boldsymbol{U}_\infty$ when ShP is the perturbed species. For all the examples $\gamma = 0.001\mathrm{s}^{-1}$ and $h/y = 0.005\mathrm{s}^{-1}$. To make the results easier to see there are only labels on the key subnetwork species.

6.2.5.2 Timescale separation

There is a difference in timescale between protein interactions and gene regulation as discussed in Section 6.1. This can be seen by looking at the the various contributions to $\boldsymbol{\chi}$ and $\boldsymbol{U}$; however the timescale separation is not always visible when analysing the concentration timecourses themselves as found from solving the linear projected equations.

Fig. 6.11 shows that if we perturb Shc by applying rates of $\gamma = 0.01\text{s}^{-1}$ and $h/y = 0.001\text{s}^{-1}$ then all the contributions from $\boldsymbol{\chi}(t')$ show timescale separation, i.e. well defined and distinct "phases" of evolution on proteomic and gene regulation timescales, respectively. Similarly if we look at the contributions from $\boldsymbol{U}$ we also see timescale separation as shown in Fig. 6.12. However this separation is not necessarily seen when we look at the timecourses found by solving the projected equations, or equivalently carrying out the convolution (6.2.13). Fig. 6.13 shows that the timescale separation is visible in the time course of RP but not the one of Grb. Looking at the expressions for the time courses written in terms of exponentials of time, one finds that both time courses have contributions with rates of $\mathcal{O}(\gamma)$, thus varying only on regulation timescales, and contributions with proteomic rates of $\mathcal{O}(\text{s}^{-1})$. In the case of RP these fast contributions have prefactors that are large enough for us to see the faster protein timescale. Most other subnetwork species have fast contributions of small amplitude, and therefore when we look at their time courses we only see the effects on the gene regulation timescale.

6.2.6 Building memory functions from response functions

Response functions are obtained by perturbing a bulk species and looking at the effect this has on the boundary species. We digress briefly to ask if there is a way to represent memory functions as superpositions of response functions. This would be intuitively plausible as memory consists of an initial perturbation in the subnetwork propagating out into the bulk, and then back from there to the subnetwork.

Let us consider a baseline system where there is a destruction rate γ_i acting on every bulk species δx_i that directly interacts with the boundary species we are interested in. If we take that boundary species and look at the corresponding response functions obtained

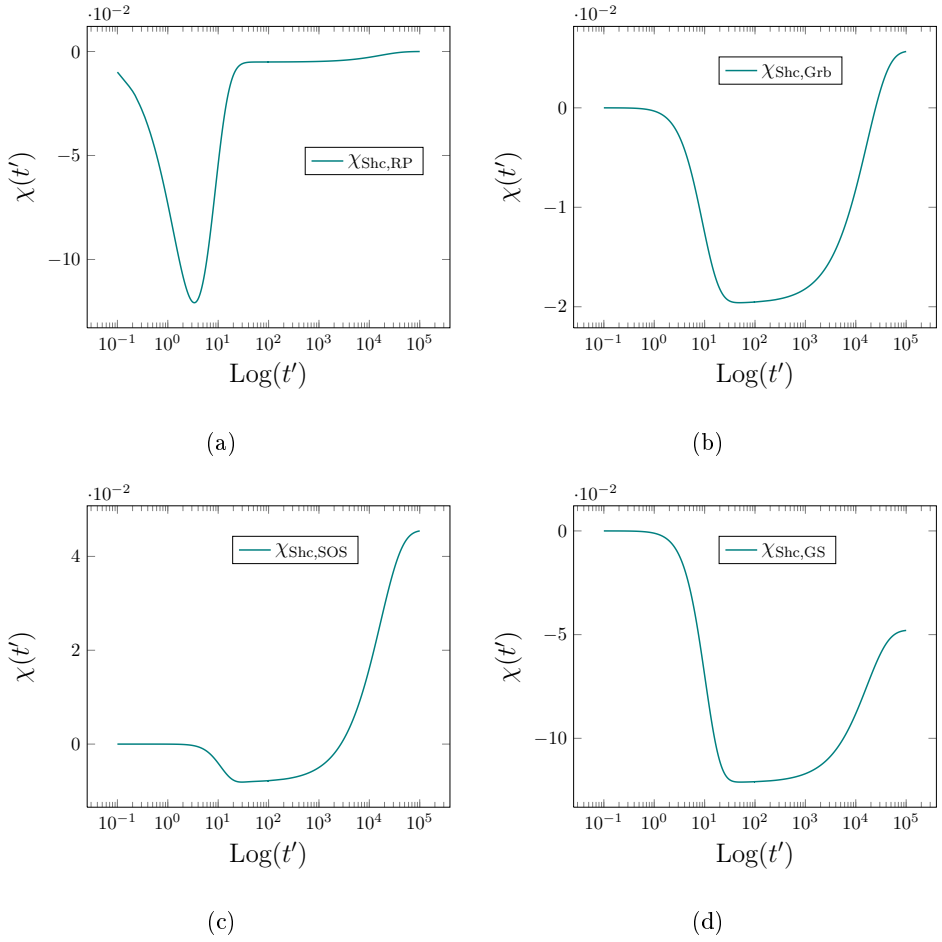


Figure 6.11: Log plots of the $\boldsymbol{\chi}$ contributions for all the boundary species – see legends – when Shc is perturbed. All the plots show timescale separation.

by perturbing each of its bulk reaction partners we can build up the memory function relating to that boundary species. The memory function is written in terms of blocks of $\boldsymbol{L}$ as (5.2.5)

$$\boldsymbol{M}(\Delta t) = \boldsymbol{L}^{\text{s,b}} e^{\boldsymbol{L}^{\text{b,b}}\Delta t} \boldsymbol{L}^{\text{b,s}} \tag{6.2.34}$$

Using the definition of $\boldsymbol{R}$ in (6.2.7) we can then write the memory function in terms of response coefficients as

$$\boldsymbol{M}(\Delta t) = \boldsymbol{L}^{\text{s,b}} \boldsymbol{R}(\Delta t) \tag{6.2.35}$$

Intuitively, then, any memory function is a linear combination of appropriate response functions.

Going back to the example of EGFR, consider the boundary species SOS. SOS interacts

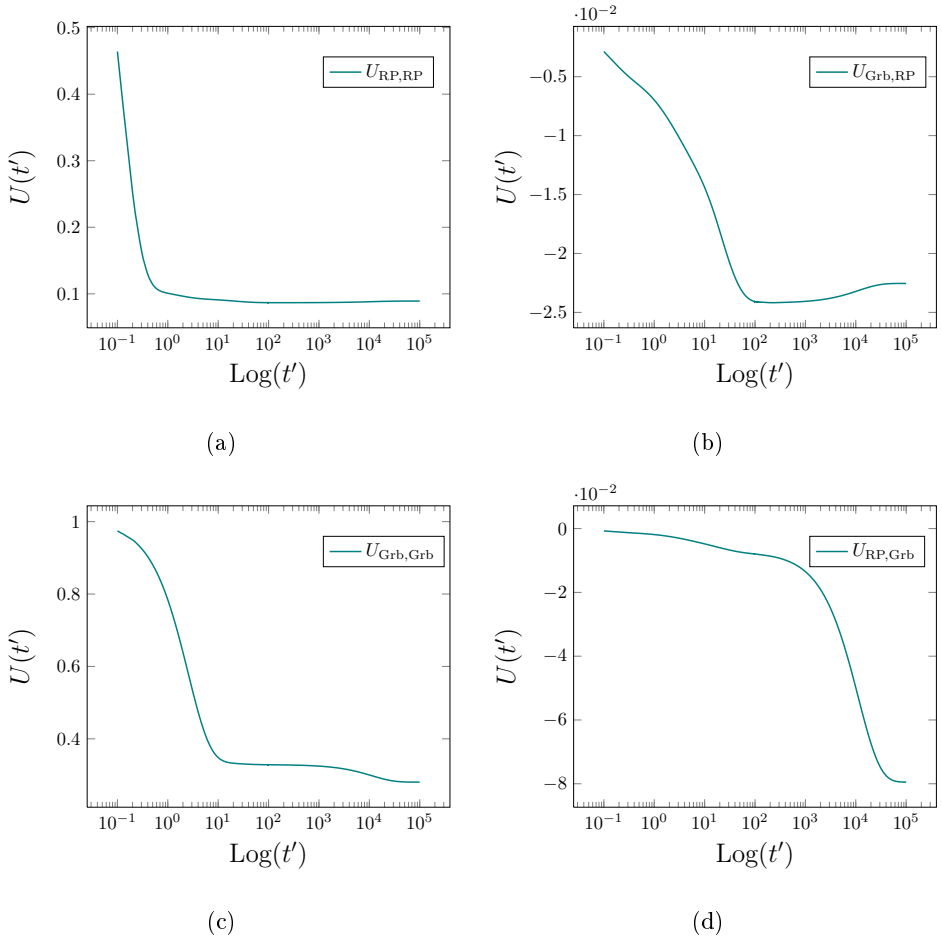


Figure 6.12: Log plots of a subset of the $\boldsymbol{U}$ contributions from the boundary species needed to obtain the responses of RP and Grb.

with ShP-Grb2 (ShG) to create Shc-Grb2-SOS (ShGS) and with RP-Shc-Grb2 (RShG) to create RP-Shc-Grb2-SOS (RSHGS). The rate of complex formation for the reaction with ShP-Grb2 (ShG) is $0.03\mathrm{nMol}^{-1}\mathrm{s}^{-1}$ and the rate of complex formation for the reaction with RP-Shc-Grb2 (RShG) is $0.01\mathrm{nMol}^{-1}\mathrm{s}^{-1}$. Considering the reaction rates and response functions from perturbing each of these bulk species we can obtain the self memory function of SOS as

$$\begin{aligned} M_{\mathrm{SOS,SOS}}(\Delta t) = {} & 0.03 y_{\mathrm{ShG}} R_{\mathrm{ShGS,SOS}}(\Delta t) - 0.03 y_{\mathrm{ShG}} R_{\mathrm{ShG,SOS}}(\Delta t) \\ & + 0.01 y_{\mathrm{RShG}} R_{\mathrm{RShGS,SOS}}(\Delta t) - 0.01 y_{\mathrm{RShG}} R_{\mathrm{RShG,SOS}}(\Delta t) \end{aligned} \tag{6.2.36}$$

Figure 6.14 confirms that the self memory function of SOS found from response functions in the manner above is the same as the original memory function. It would be interesting to investigate further whether there is a connection to the Fluctuation Dissipation

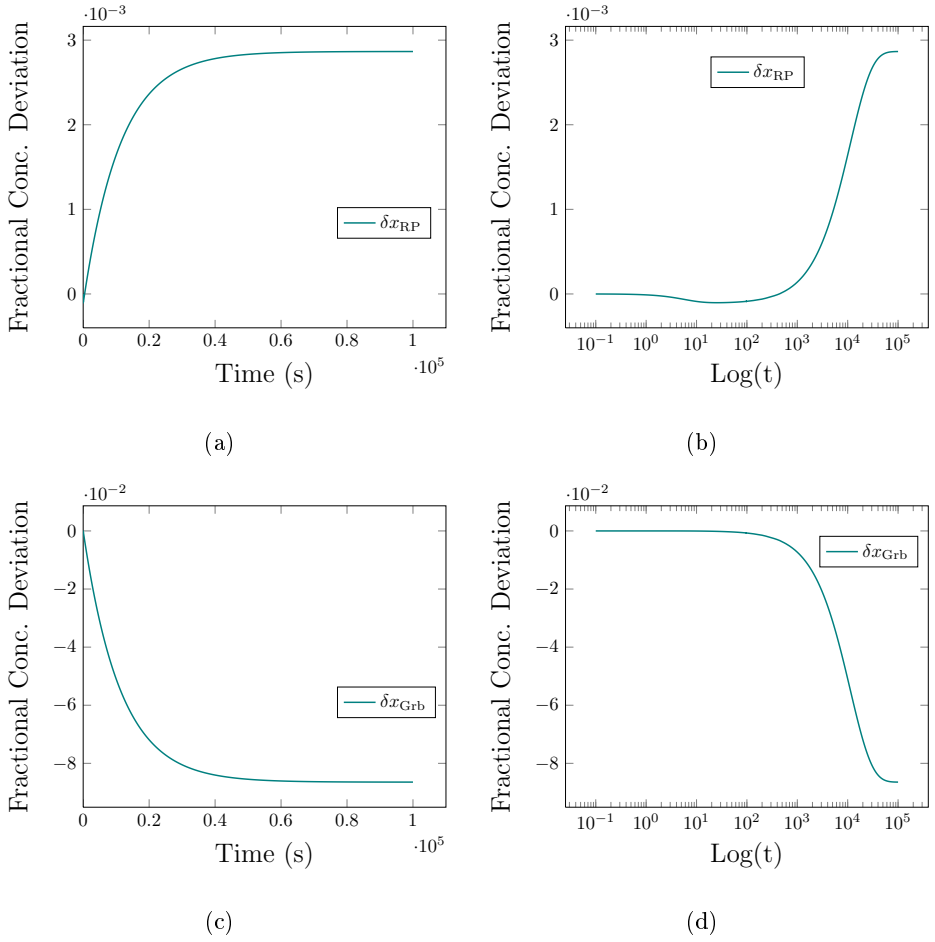


Figure 6.13: Plots of time courses for phosphorylated EGFR (RP) and Grb2 on a normal time scale and a log time scale. The log plots show timescale separation for RP but not for Grb2. $\gamma = 0.01\mathrm{s}^{-1}$ and $h/y = 0.001\mathrm{s}^{-1}$.

Theorem. One could start by looking at systems where there is detailed balance as then the memory function is the correlation function of the random force, so that (6.2.35) has the typical FDT character of relating a correlation function to a response.

6.3 Nonlinear response

Analysing linear response allows us to understand how bulk perturbations influence the subnetwork dynamics, but the results are only accurate for small values of h/γ (see Fig. 6.15). We would also like to be able to model experiments such as gene knockdowns where the change in concentrations of the perturbed bulk species is larger and so we

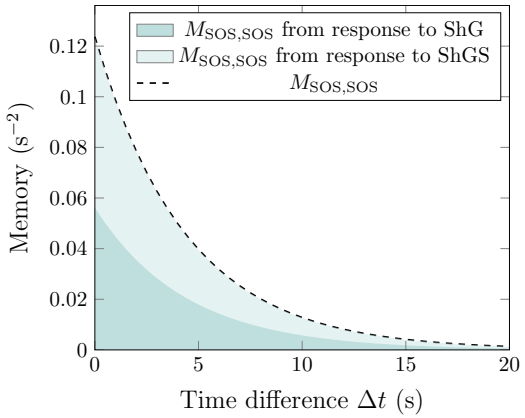


Figure 6.14: The self memory function of SOS found from summing up the response functions relating to its bulk interaction partners compared to the original self memory function of SOS. The two main contributions are from the perturbation of ShP-Grb2 (ShG) and Shc-Grb2-SOS (ShGS).

need to be able to analyse nonlinear responses.

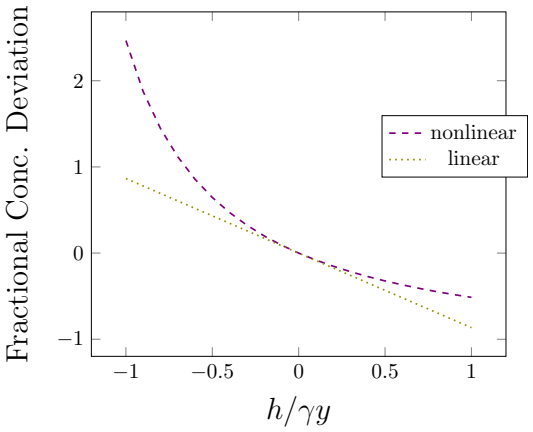


Figure 6.15: The solutions to the linearised and nonlinear reaction equations for Grb2, after perturbing Shc, plotted against $h_{\mathrm{Shc}}/\gamma_{\mathrm{Shc}}y_{\mathrm{Shc}}$. With a fixed γ of $0.01\mathrm{s}^{-1}$.

To find the form of the nonlinear projected equations we use the perturbed exponential operator (6.2.1) and then calculate the memory function in the same way as in Section 6.2. We will include all terms up to quadratic order in δx as well as terms of $\mathcal{O}(h)$ and $\mathcal{O}(h\delta x)$. This is due to the fact that we expect $\delta x \sim \mathcal{O}(h/\gamma)$ (see (6.2.32)) and as we are considering nonlinear dynamics we want to keep all terms in the projected equations up to $\mathcal{O}(\delta x^2)$. With the scaling of $\delta x \sim \mathcal{O}(h/\gamma)$ then terms of $h\delta x \sim \mathcal{O}(\delta x^2\gamma)$, which is small compared to δx^2. We nevertheless keep these terms, as in the linearised dynamics

one similarly needs to keep both $\mathcal{O}(\delta x)$ and $\mathcal{O}(h) = \mathcal{O}(\delta x\, \gamma)$. However, we discard terms of $\mathcal{O}(h^2) = \mathcal{O}(\delta x^2\, \gamma^2)$ as these are even smaller and should be irrelevant compared to both the leading terms of $\mathcal{O}(\delta x^2)$ and the first subleading $\mathcal{O}(\delta x^2\, \gamma)$ corrections.

We first look at the perturbation on the memory function as defined in equation (5.2.5). As before we consider only bulk perturbations and therefore $\delta\mathcal{L}a_i = 0$ whenever a_i is any subnetwork concentration or product of subnetwork concentrations. Applying the operators defining the memory function to such a subnetwork observable we thus find

$$\begin{aligned}\mathcal{P}\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}\mathcal{Q}\Delta t}\mathcal{Q}\mathcal{L}a_i = \mathcal{P}\mathcal{L}_0\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}\Delta t}\mathcal{Q}\mathcal{L}_0 a_i + \mathcal{P}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}\Delta t}\mathcal{Q}\mathcal{L}_0 a_i \\ + \mathcal{P}\mathcal{L}_0\mathcal{Q}\int_0^{\Delta t} dt''\, e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t''}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(\Delta t - t'')}\mathcal{Q}\mathcal{L}_0 a_i \\ + \mathcal{P}\delta\mathcal{L}\mathcal{Q}\int_0^{\Delta t} dt''\, e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t''}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(\Delta t - t'')}\mathcal{Q}\mathcal{L}_0 a_i\end{aligned} \tag{6.3.1}$$

where we have inserted $\mathcal{L} = \mathcal{L}_0 + \delta\mathcal{L}$ and exploited the fact that $\mathcal{L}a_i = \mathcal{L}_0 a_i$. The first term of (6.3.1) is the original memory function without perturbations and the last term contains two $\delta\mathcal{L}$ operators. It will then contribute only at $\mathcal{O}(h^2)$ and we will discard it as explained above.

To evaluate the second and third term of (6.3.1), one notes that where $\delta\mathcal{L}$ acts on linear observables resulting from the operators on its right being applied to a_i, it produces zero or constants. These do not contribute to the memory function by the same arguments as given for linearised dynamics in Sec. 6.2.

When we apply $\delta\mathcal{L}$ to quadratic observables, on the other hand, it will map these to linear observables. Overall, and assuming that third order terms on δx can be ignored, $\delta\mathcal{L}$ can then be represented as a matrix of the following form:

$$\boldsymbol{H} = \begin{pmatrix} 0 & 0 & 0 & \boldsymbol{H}^{\mathrm{s,sb}} & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{H}^{\mathrm{b,bb}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.3.2}$$

where the nonzero entries are all $\mathcal{O}(h)$.

Now let $\boldsymbol{z}^{\mathrm{T}} = (\boldsymbol{\delta x}, \boldsymbol{\delta x}^2)^{\mathrm{T}}$ as before, then writing the middle two terms of (6.3.1) in the form of matrix operators we find that the first $\mathcal{O}(h)$ contribution to the memory

function is

$$\mathcal{P}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}\Delta t}\mathcal{Q}\mathcal{L}_0 z_i = \left[\boldsymbol{z}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{H}\boldsymbol{Q}e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}\Delta t}\boldsymbol{Q}\boldsymbol{L}_0\right]_i \tag{6.3.3}$$

and the second $\mathcal{O}(h)$ contribution, from the expansion of the operator exponential, is

$$\begin{aligned}\mathcal{P}\mathcal{L}_0\mathcal{Q}\int_0^{\Delta t} dt''\, e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t''}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(\Delta t-t'')}\mathcal{Q}\mathcal{L}_0 z_i \\ = \left[\boldsymbol{z}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{L}_o\boldsymbol{Q}\int_0^{\Delta t} dt''\, e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}t''}\boldsymbol{Q}\boldsymbol{H}\boldsymbol{Q}e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}(\Delta t-t'')}\boldsymbol{Q}\boldsymbol{L}_0\right]_i\end{aligned} \tag{6.3.4}$$

Together these two terms give us the $\mathcal{O}(h)$ contributions to the memory function.

When we insert the perturbed memory functions into the projected equations (5.2.3) they are multiplied by past concentrations fluctuations, i.e. $\mathcal{O}(\delta x)$, or products of such fluctuations, $\mathcal{O}(\delta x^2)$. To be systematic we therefore only include the $\mathcal{O}(h)$ correction of the memory function in terms of the first type (linear memory), but discard it for the second type (quadratic memory) where it would constitute a contribution of $\mathcal{O}(h\delta x^2) = \mathcal{O}(\delta x^3\,\gamma^2)$. It is also worth pointing out that the h-dependent terms in the memory functions do not change the boundary structure, i.e. it is still only the boundary species in the subnetwork that contain memory terms in their time evolution equations.

We can provide a similar analysis to obtain the extra contributions to the random force. Using the exponential expansion (6.2.1) we find

$$e^{\mathcal{Q}\mathcal{L}\mathcal{Q}t}\mathcal{Q}\mathcal{L}a_i = e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t}\mathcal{Q}\mathcal{L}_0 a_i + \int_0^t dt'\, e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')}\mathcal{Q}\mathcal{L}_0 a_i \tag{6.3.5}$$

where the first term is the original random force contribution. The second term of (6.3.5) is written using matrix operators as

$$\begin{aligned}\int_0^t dt'\, e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}t'}\mathcal{Q}\delta\mathcal{L}\mathcal{Q}e^{\mathcal{Q}\mathcal{L}_0\mathcal{Q}(t-t')}\mathcal{Q}\mathcal{L}_0 a_i = \int_0^t dt'\, \left[\boldsymbol{h}^{\mathrm{T}}e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}(t-t')}\boldsymbol{Q}\boldsymbol{L}_0\right]_i \\ + \left[\boldsymbol{z}^{\mathrm{T}}\int_0^t dt'\, e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}t'}\boldsymbol{Q}\boldsymbol{H}\boldsymbol{Q}e^{\boldsymbol{Q}\boldsymbol{L}_0\boldsymbol{Q}(t-t')}\boldsymbol{Q}\boldsymbol{L}_0\right]_i\end{aligned} \tag{6.3.6}$$

Here the first term of (6.3.6) is the contribution from the linear observables that arise to the right of $\delta\mathcal{L}$, and is explained in Sec. 6.2. In the second term we retain only the terms of $\mathcal{O}(h\delta x)$, i.e. discard the ones that result from the quadratic components of $\boldsymbol{z}$. We now have all the elements for describing the full projected subnetwork dynamics of a perturbed system.

6.3.1 New steady state

The steady state of the nonlinear perturbed projected equations cannot be found directly by splitting into transient and persistent contributions in the same way as for the linear response. Instead we directly solve the projected equations in the long time limit. Let $\boldsymbol{z}^{\mathrm{s}^{\mathrm{T}}} = (\boldsymbol{\delta x}^{\mathrm{s}}, \boldsymbol{\delta x}^{\mathrm{s}^2})^{\mathrm{T}}$ collect the linear and quadratic subnetwork observables, then we can write the steady state condition as

$$0 = \boldsymbol{z}^{\mathrm{s}^{\mathrm{T}}} \boldsymbol{\Omega} + \boldsymbol{z}^{\mathrm{s}^{\mathrm{T}}} \overline{\boldsymbol{M}} + \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\chi} \tag{6.3.7}$$

where $\overline{\boldsymbol{M}}$ also includes terms of $\mathcal{O}(h)$ and $\boldsymbol{\chi}$ is the integrated response function as used for the linearised dynamics. We can then solve these equations numerically to find the nonlinear perturbed steady state $\boldsymbol{\delta x}^{\mathrm{s}}$. The matrices $\boldsymbol{\Omega}$ and $\overline{\boldsymbol{M}}$ contain nonlinear contributions that mean it is not obvious whether there is a nonlinear analogue of the linear boundary structure, i.e. the decomposition into $\boldsymbol{\chi}$ and $\boldsymbol{U}$. This question would merit further study. One could also look in more detail at how significant the $\mathcal{O}(h)$ contribution in $\overline{\boldsymbol{M}}$ is, i.e. how much the predicted steady state is affected by these terms. It should be emphasised that although we are only calculating the projected equations to $\mathcal{O}(h)$ and $\mathcal{O}(h\delta x)$ in h, when we solve the nonlinear steady state conditions we do obtain solutions that in principle involve all orders of h.

For a large subnetwork there will be a substantial set of nonlinear equations to solve; one way to tackle this is to proceed by iteration. We know that for small perturbations, i.e. small values of $h/\gamma y$, the steady state found from linear response must be close to the result from nonlinear response. Therefore for larger systems we can calculate the linear response for a small perturbation and then use this result as the starting point for our iteration to obtain the required nonlinear response. Typically we proceed by considering values of h increasing or decreasing away from zero, so that the continuity of the solution with h can be exploited.

Fig. 6.16 illustrates for some of the species from the EGFR network that the linear response is only accurate for relatively small perturbations when compared to a direct solution of the full (subnetwork and bulk) system of steady state conditions. The figure also shows that the nonlinear perturbed projected equations are more accurate for positive perturbations and show somewhat larger deviations for negative perturbations,

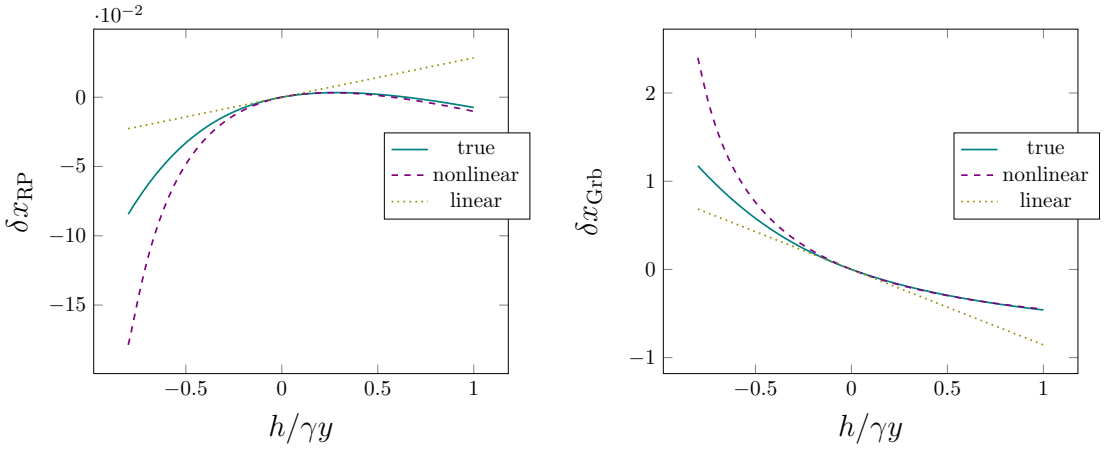


Figure 6.16: Steady state response vs. the size of the perturbation for the true reaction equations and the nonlinear and linear projected equations. We have perturbed Shc in the bulk with a degradation rate $\gamma = 0.01\mathrm{s}^{-1}$ and varying values of h. The plots show how this affects the steady state of (a) phosphorylated RP; (b)Grb2.

i.e. knockdowns, which have a comparatively larger effect on the steady state. We notice also that in Fig. 6.16(a) when the value of $h/\gamma y$ is large the sign of the nonlinear response can be different to the sign of the linear response, and this qualitative effect is well captured by the nonlinear projected equations.

Finally, we can also use the projected equations to obtain the time courses of the nonlinear response to a perturbation. To solve the equations we write them as a set of differential equations using an extended version of the method for solving the projected equations without perturbations (see Appendix B). Fig. 6.17 shows for the example in Fig. 6.2(c) that the nonlinear projected equations are a more accurate representation of the dynamics and provide a substantial improvement in accuracy for large perturbations, whereas as expected for smaller perturbations the differences are modest. We do not include time courses for EGFR here due to numerical complications that we encountered in solving the full nonlinear projected equations for this larger system.

6.3.1.1 Including copy number fluctuations

In this chapter I have used the projection method to analyse the response of protein interaction networks to perturbations. The results have been compared to deterministic

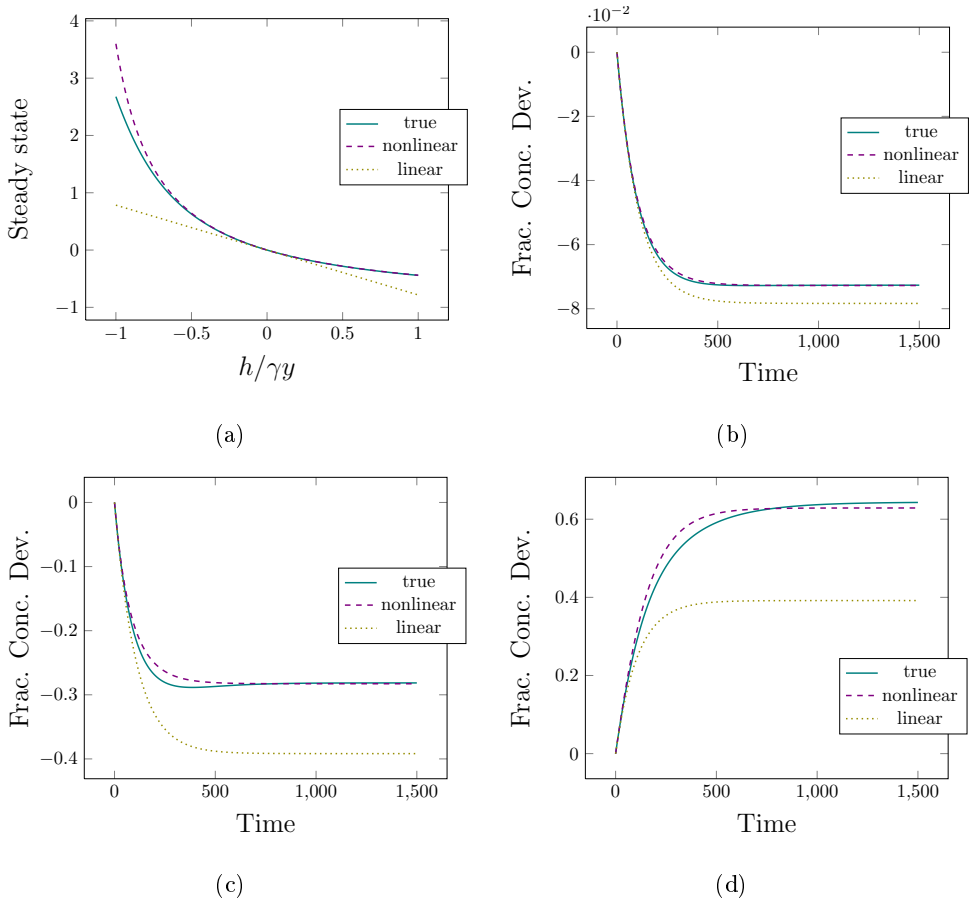


Figure 6.17: Steady state and timecourses for δx_1 in the example in Fig. 6.2(c) for the true reaction equations and the nonlinear and linear projected equations when δx_4 is being perturbed. (a) Steady state vs. perturbation; (b) $h/\gamma y = 0.1$; (c) $h/\gamma y = 0.5$; (d) $h/\gamma y = -0.5$.

reaction equations. However, it is also possible to compare the results to chemical Langevin equations where the account for a finite level of copy number noise rather than taking $\epsilon \to 0$ as we have done so far. Fig. 6.18 shows a solution of the Langevin equations for δx_1 and δx_2 for the example shown in Fig. 6.2(c) with different values of ϵ, for the case when a perturbation of $h/\gamma y = 0.1$ was applied to δx_4. As expected, as ϵ increases the amount of variation in the results also increases.

To assess systematic effects from the presence of copy number noise, for each value of ϵ we averaged over three runs and looked at the difference between this finite-noise steady state and the steady state found from the noise-free projected equations. Figure 6.19 shows that as the size of ϵ increases the deviation between the steady state predicted by the projected equations and the "true" mean steady state of the chemical Langevin equations also increases. This is again as expected, and applies to the steady state of both δx_1 and δx_2. More importantly, we see that the ϵ-dependence of the steady state is rather weak in the example shown, and of the order of the difference in predictions between the noise free reaction equations and the projected equations. Thus to a first approximation we can say that moderate copy number noise does not qualitatively change the accuracy of the projected equations, although clearly this working hypothesis should be checked across a range of initial conditions, perturbation strengths and indeed protein interaction network structures.

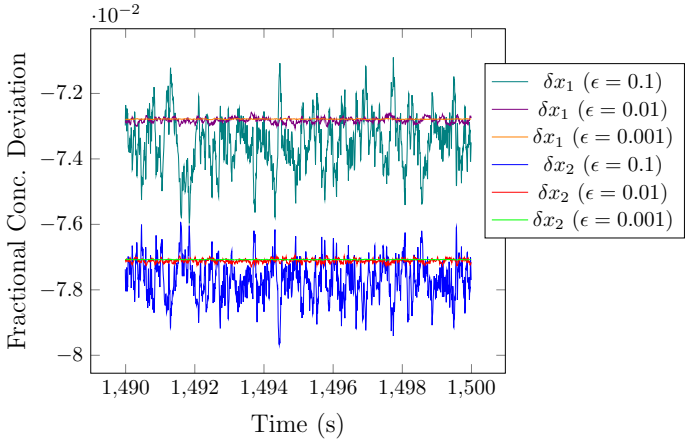


Figure 6.18: The timecourses for δx_1 and δx_2 in the example in Fig. 6.2(c) for the chemical Langevin equations when δx_4 is being perturbed with a perturbation of $h/\gamma y = 0.1$.

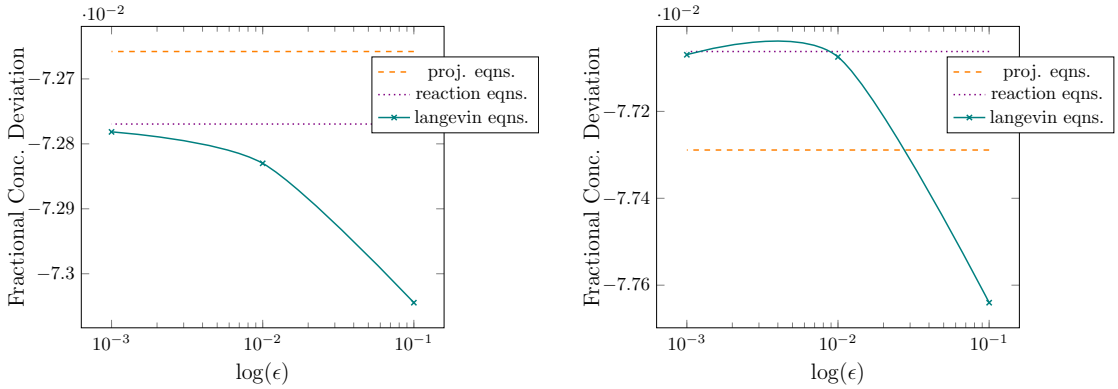


Figure 6.19: The difference between the steady state solutions to the projected equations, reaction equations and (on average) chemical Langevin equations with a perturbation of $h/\gamma y = 0.1$, for (a) δx_1 and (b) δx_2. Lines for the mean Langevin equation solution are only guides to the eye.

6.4 Summary

We have extended the projection approach to also include perturbations, which were modelled as simple creation and destruction terms in the equations as might be expected due to gene regulation processes. We first considered linear response where we found that we can split the response into two parts: the propagation of the perturbation through the bulk and on to the boundary, and the response to the boundary perturbation throughout the subnetwork. These two terms can then each be split into transient and persistent parts. We have also shown that we can use the projected equations to describe nonlinear response, and obtain more accurate results. This makes the analysis more applicable to experimental situations, where large perturbations such as knockdowns are the rule rather than the exception.

Chapter 7

Random force statistics

When modelling a subnetwork embedded in a bulk network it is widely acknowledged that the presence of a surrounding bulk system generates extrinsic noise on the subnetwork dynamics [9, 10]. The projected equations describe the evolution of conditional averages where a component of the extrinsic noise that affects the conditional averages comes from the bulk initial conditions. This is represented by the random force where every boundary species has a random force contribution. The random force can then be used to try and understand the statistical properties of this component of the extrinsic noise. We also want to analyse the accuracy of the projected equations when we have a random force that is not in steady state.

In this Chapter I will begin our analysis of the properties of the random force. In Section 7.1, I look at the accuracy of the linearised projected equations. I then analyse different approximations that can be used to represent the random force in Sec. 7.2. Finally, in Sec. 7.3, I look at the nonlinear dynamics and the accuracy of the equations with different approximations of the random force.

7.1 The accuracy of the projected equations

To analyse the projected equations we have previously assumed that the bulk is at steady state and therefore the random force is set to zero. We want to look at how accurate our results will be if the bulk is not at steady state. We start with the linearised dynamics

and sample bulk initial conditions from Gaussian distributions centered around zero with a variance equal to $\langle \delta x^2 \rangle = \epsilon y$. This is the small fluctuation limit of copy number fluctuations for each individual bulk species following Poisson statistics. We fix the subnetwork initial conditions and then solve both the reaction equations and projected equations using the sampled nonzero bulk initial conditions. Repeating this process, we get at each point in time a distribution of predicted (conditionally averaged) subnetwork concentrations, one according to the projected equations and one according to the full reaction equations. Our goal is to assess how close these two distributions are, i.e. to quantify how well the projected equations capture fluctuations due to variation in bulk initial conditions.

One possible measure of similarity between the true distributions $p(\delta x)$ and $q(\delta x)$ is the Kullback-Leiber (KL) distance [73]. For the linearised dynamics all distributions are Gaussian, and if we focus on the distribution of a single concentration then the KL divergence between two univariate Gaussian distributions, the true distribution (full reaction equations) $p(\delta x) = N(\mu_1, \sigma_1^2)$ and its approximation (projected equations) $q(\delta x) = N(\mu_2, \sigma_2^2)$ is

$$KL(p,q) = \frac{1}{2} \ln \frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \tag{7.1.1}$$

To evaluate this we then just need to collect the relevant means and variances.

Another measure that can be used to look at the accuracy of the projected equations is the normalised mean square error of the correlation matrices

$$\text{error} = \frac{\sum_{i,j} \left(C_{ij} - \hat{C}_{ij} \right)^2}{\sum_{i,j} C_{ij}^2} \tag{7.1.2}$$

where C_{ij} is the correlation matrix of the concentration fluctuations predicted by the reaction equations and $\hat{C}_{ij}$ the correlation matrix from the projected equations. This measure has the advantage of picking up also the accuracy of correlations in the fluctuations between different concentrations, while the single-variable KL is insensitive to these.

Firstly we consider the model network from Fig. 7.1, and use the single-variable KL divergences to quantify the accuracy of the projected equations. We sampled 200 bulk initial conditions and then calculated the KL between the distributions predicted by the

linearised reaction equations and the linearised projected equations. In the latter, the full time-dependence of the random force from (3.1.10) is accounted for. Fig. 7.2 shows that the projected equations are very accurate when we include the random force, with the KL divergence being close to zero. Indeed for linearised dynamics the projected equations should be *exact* once the random force is included, so we would expect the KL divergence approach zero as the sample size increased.

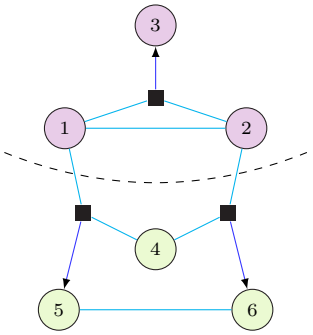


Figure 7.1: Sketch of example protein interaction network.

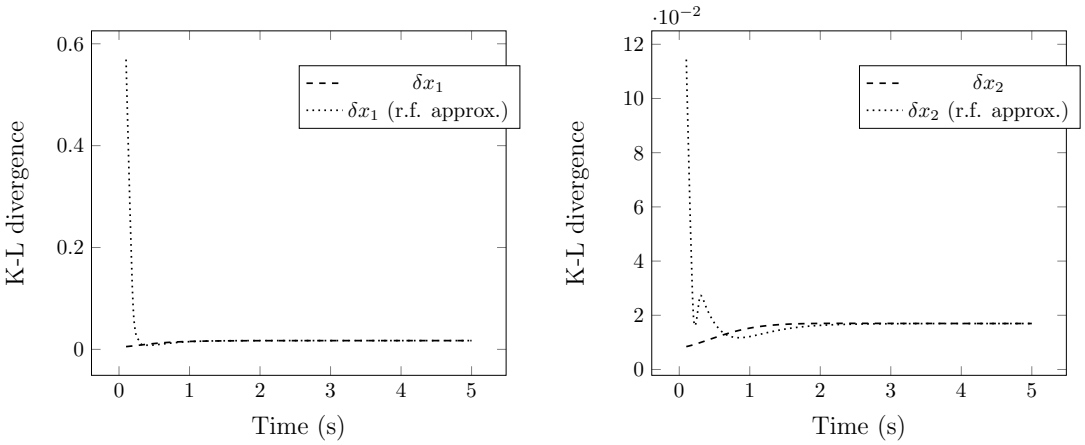
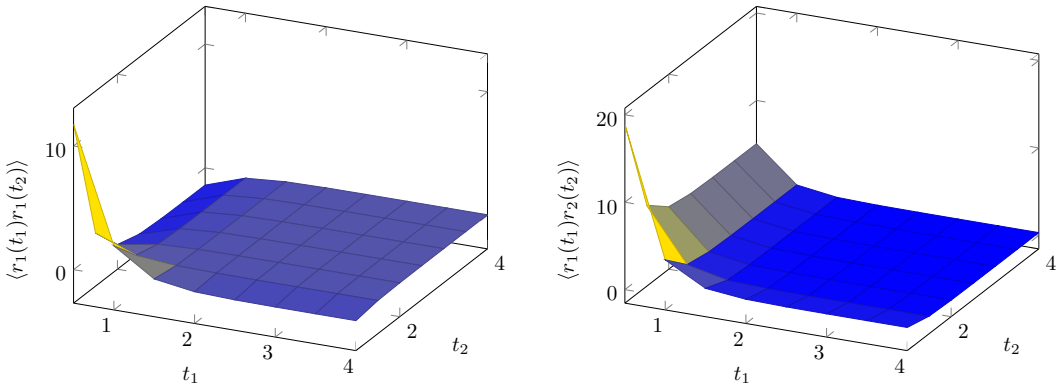


Figure 7.2: The KL divergence for the example in Fig. 7.1, comparing the linearised reaction equations with the linearised projected equations with the full random force; the KL values are small and are nonzero only because of the finite number of samples taken. Dotted: results when the random force is approximated as an initial kick plus a constant. Shown are the KL divergences between the projected and real distributions of (a) $\delta x_1(t)$; (b) $\delta x_2(t)$.

7.2 Approximating the random force

In practice we do not want to have to explicitly calculate $\boldsymbol{r}$ as a function of time. We therefore want to see if there are simple ways of approximating the random force. To help us understand what the statistics of the random force are, Fig. 7.3 shows a self-correlation and a cross-correlation function between random forces. One sees that the random forces in general have constant pieces for large times, as well as short timescale contributions. Our aim is to propose a simple approximation that can capture both features. An initial idea would to focus on getting the constant piece of the random



(a) (b)

Figure 7.3: Correlators of the random force for the example network in Fig. 7.1 show short-time variation as well as constant parts for longer times, though quantitatively the latter are quite small.

force correct as that might be expected to be dominant in determining the random force-dependent perturbation of the steady state concentrations. This approximation would amount to replacing $\boldsymbol{r}(t)$ by $\boldsymbol{r}_\infty = \lim_{t\to\infty} \boldsymbol{r}(t)$. We will see below, however, that this is too naive to give accurate approximations to the random force effects on the subnetwork concentrations.

To understand in more detail what properties the approximated random force should have we go back to the projected equations, which are written

$$\frac{\partial}{\partial t} a_\alpha(t) = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \int_0^t dt' \sum_\beta a_\beta(t')M_{\beta\alpha}(t-t') + r_\alpha(t) \tag{7.2.1}$$

In the current case of linearised dynamics, we use for the a_α the concentration (devi-

ations) in the subnetwork as before, denoted collectively by $\boldsymbol{\delta x}^{\mathrm{s}}$. To write down the solution of the projected equations in the presence of nonzero random forces, we can exploit a helpful analogy with the treatment of external (bulk) perturbations in Chapter 6, where we introduced the Green's function $\boldsymbol{U}(t)$ for the unperturbed system, which is characterized by the rate matrix and memory function. In terms of this Green's function, the subnetwork concentrations at time t are simply

$$\boldsymbol{\delta x}^{\mathrm{s}^{\mathrm{T}}}(t) = \int_0^t dt'\, \boldsymbol{r}^{\mathrm{T}}(t')\boldsymbol{U}(t-t') + \boldsymbol{\delta x}^{\mathrm{s}^{\mathrm{T}}}(0)\boldsymbol{U}(t) \tag{7.2.2}$$

The second term depends only on the subnetwork initial conditions and not the random force, so we will ignore it in the following discussion. Taking a Laplace transform of the random force-dependent term, this reads

$$\boldsymbol{\delta}\hat{\boldsymbol{x}}^{\mathrm{s}^{\mathrm{T}}}(z) = \hat{\boldsymbol{r}}(z)^{\mathrm{T}}\hat{\boldsymbol{U}}(z) \tag{7.2.3}$$

where $\hat{\boldsymbol{r}}(z)$ is the Laplace Transform of the random force. Decomposing $\hat{\boldsymbol{U}}(z)$ and $\hat{\boldsymbol{r}}(z)$ as we did in our analysis of response to perturbations in Chapter 6, we write

$$\begin{aligned} \hat{\boldsymbol{r}}(z) &= \frac{1}{z}\boldsymbol{r}_\infty(z) + \Delta\hat{\boldsymbol{r}}(z) \\ \hat{\boldsymbol{U}}(z) &= \frac{1}{z}\boldsymbol{U}_\infty + \Delta\hat{\boldsymbol{U}}(z) \end{aligned} \tag{7.2.4}$$

The perturbation of the steady state due to the random force is then found as

$$\begin{aligned} \boldsymbol{\delta x}^{\mathrm{s}}(t\to\infty) &= \lim_{z\to 0} z\hat{\boldsymbol{r}}^{\mathrm{T}}(z)\hat{\boldsymbol{U}}(z) \\ &= \lim_{z\to 0}\left(\frac{1}{z}\boldsymbol{r}_\infty(z) + \Delta\hat{\boldsymbol{r}}(z)\right)^{\mathrm{T}}\left(\frac{1}{z}\boldsymbol{U}_\infty + \Delta\hat{\boldsymbol{U}}(z)\right) \\ &= \lim_{z\to 0}\frac{1}{z}\boldsymbol{r}_\infty^{\mathrm{T}}\boldsymbol{U}_\infty + \boldsymbol{r}_\infty^{\mathrm{T}}\Delta\hat{\boldsymbol{U}}(z) + \Delta\hat{\boldsymbol{r}}^{\mathrm{T}}(z)\boldsymbol{U}_\infty + z\Delta\hat{\boldsymbol{r}}^{\mathrm{T}}(z)\Delta\hat{\boldsymbol{U}}(z)\mathcal{O}(z) \end{aligned} \tag{7.2.5}$$

We show below that the apparently divergent first term does not contribute; the last term vanishes for $z\to 0$. In the two remaining finite contributions, the small z-limit gives

$$\Delta\hat{\boldsymbol{r}}(z\to 0) = \int_0^\infty dt'\,\Delta\boldsymbol{r}(t') = \int_0^\infty dt'\,[\boldsymbol{r}(t') - \boldsymbol{r}_\infty] \equiv \overline{\Delta\boldsymbol{r}} \tag{7.2.6}$$

and similarly $\Delta\hat{\boldsymbol{U}}(z\to 0) = \overline{\Delta\boldsymbol{U}}$, so that

$$\boldsymbol{\delta x}^{\mathrm{s}}(t\to\infty) = \boldsymbol{r}_\infty^{\mathrm{T}}\overline{\Delta\boldsymbol{U}} + \overline{\Delta\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{U}_\infty \tag{7.2.7}$$

This decomposition is directly analogous to equation (6.2.17) in Chapter 6, with the contributions being respectively the transient response to the persistent part of the random force, and the persistent response to the transient part of the random force.

The analysis so far tells us that we will get the correct random force-dependent steady state from an approximate form of the random force as long as the approximation preserves both $\boldsymbol{r}_\infty$ – as in the naive approximation discussed above – and $\overline{\Delta \boldsymbol{r}}$. Our proposal is then to construct the approximation by "concentrating" the transient part of the random force into a spike at $t = 0$, specifically

$$\boldsymbol{r}(t) = \boldsymbol{r}_\infty + \Delta \boldsymbol{r}(t) \simeq \boldsymbol{r}_\infty + \delta(t)\overline{\Delta \boldsymbol{r}} \tag{7.2.8}$$

Accordingly we will then approximate the linearised projected equations as

$$\frac{\partial}{\partial t}\boldsymbol{\delta x}^{\mathrm{s}^{\mathrm{T}}} = \boldsymbol{\delta x}^{\mathrm{s}^{\mathrm{T}}}\boldsymbol{\Omega} + \int_0^t dt'\, \boldsymbol{\delta x}^{\mathrm{s}^{\mathrm{T}}}\boldsymbol{M}(t-t') + \boldsymbol{r}_\infty + \overline{\Delta \boldsymbol{r}}\delta(t) \tag{7.2.9}$$

Integrating a small time interval around $t = 0$ then shows that the effect of the δ-piece of the random force is equivalent to a random change in the initial subnetwork conditions, by an amount $\overline{\Delta \boldsymbol{r}}$.

To see how the above approximation is implemented in practice, we recall that in terms of the subblocks of the matrix $\boldsymbol{L}$ representing the adjoint Fokker-Planck matrix operator the random force is written as

$$\boldsymbol{r}^{\mathrm{T}}(t) = \boldsymbol{\delta x}^{\mathrm{b}}(0)^{\mathrm{T}} e^{\boldsymbol{L}^{\mathrm{b,b}}t}\boldsymbol{L}^{\mathrm{b,s}} \tag{7.2.10}$$

which after Laplace transforming reads

$$\hat{\boldsymbol{r}}(z) = \boldsymbol{\delta x}^{\mathrm{b}^{\mathrm{T}}}(0)(z - \boldsymbol{L}^{\mathrm{b,b}})^{-1}\boldsymbol{L}^{\mathrm{b,s}} = \boldsymbol{\delta x}^{\mathrm{b}^{\mathrm{T}}}(0)\hat{\boldsymbol{R}}(z) \tag{7.2.11}$$

where $\hat{\boldsymbol{R}}(z)$ is the Laplace transform of the impulse response function defined in Chapter 6.

Then if $\bar{\boldsymbol{\rho}}$ and $\bar{\boldsymbol{l}}$ are the right and left eigenvectors and μ the eigenvalues of the matrix $\boldsymbol{L}^{\mathrm{b,b}}$ we can write the persistent and transient components of the random force explicitly as

$$\begin{aligned} \boldsymbol{r}_\infty &= \boldsymbol{\delta x}^{\mathrm{b}^{\mathrm{T}}}(0) \sum_{\beta,\mu_\beta=0} \bar{\boldsymbol{\rho}}_\beta \bar{\boldsymbol{l}}_\beta^{\mathrm{T}} \boldsymbol{L}^{\mathrm{b,s}} \\ \Delta \bar{\boldsymbol{r}} &= -\boldsymbol{\delta x}^{\mathrm{b}^{\mathrm{T}}}(0) \sum_{\beta,\mu_\beta\neq 0} \frac{1}{\mu_\beta} \bar{\boldsymbol{\rho}}_\beta \bar{\boldsymbol{l}}_\beta^{\mathrm{T}} \boldsymbol{L}^{\mathrm{b,s}} \end{aligned} \tag{7.2.12}$$

There are further possible simplifications to the random force approximation which would be to take into account only one of the two pieces of the random force, either

the transient or the persistent one, the latter corresponding to the naive approximation discussed above.

We can use the eigenvector structure of $\boldsymbol{r}_\infty$ and $\boldsymbol{U}_\infty$ to show that the first term in (7.2.5) vanishes. Let $\boldsymbol{\rho}^{\mathrm{s}}$ and $\boldsymbol{l}^{\mathrm{s}}$ be the subnetwork components of the right and left eigenvectors of $\boldsymbol{L}$ and λ the corresponding eigenvalues. Then we can write $\boldsymbol{r}_\infty \boldsymbol{U}_\infty$ using (7.2.12) and (6.2.30) as

$$\begin{aligned}\boldsymbol{r}_\infty \boldsymbol{U}_\infty &= \boldsymbol{\delta x}^{\mathrm{b}^{\mathrm{T}}}(0) \sum_{\beta,\mu_\beta=0} \sum_{\alpha,\lambda_\alpha=0} \bar{\boldsymbol{\rho}}_\beta \bar{\boldsymbol{l}}_\beta^{\mathrm{T}} \boldsymbol{L}^{\mathrm{b,s}} \boldsymbol{\rho}_\alpha^{\mathrm{s}} \boldsymbol{l}_\alpha^{\mathrm{s}^{\mathrm{T}}} \\ &= \boldsymbol{\delta x}^{\mathrm{b}^{\mathrm{T}}}(0) \sum_{\beta,\mu_\beta=0} \sum_{\alpha,\lambda_\alpha=0} \bar{\boldsymbol{\rho}}_\beta \bar{\boldsymbol{l}}_\beta^{\mathrm{T}} \boldsymbol{L}^{\mathrm{b,b}} \boldsymbol{\rho}_\alpha^{\mathrm{b}} \boldsymbol{l}_\alpha^{\mathrm{s}^{\mathrm{T}}} \\ &= 0\end{aligned} \tag{7.2.13}$$

where $\bar{\boldsymbol{l}}_\beta^{\mathrm{T}} \boldsymbol{L}^{\mathrm{b,b}} = 0$ because $\bar{\boldsymbol{l}}_\beta$ is an eigenvector with zero eigenvalue of $\boldsymbol{L}^{\mathrm{b,b}}$. We have also used the fact that $\boldsymbol{L}^{\mathrm{b,s}} \boldsymbol{\rho}_\alpha^{\mathrm{s}} + \boldsymbol{L}^{\mathrm{b,b}} \boldsymbol{\rho}_\alpha^{\mathrm{b}} = 0$ as in Sec. 6.2.2.1.

7.2.1 Examples

We first consider the example from Fig. 7.1 where the random force has both a constant (persistent) term and short time (transient) piece. Fig. 7.5(a) shows that in this case just taking the persistent piece of the random force is not a good approximation as the KL divergences that result are very large.

Looking at the random force for δx_1 in the example for a range of different initial conditions (see Fig. 7.4) we see that the constant pieces of the random force are very small. This clarifies why we cannot approximate the random force purely as a constant, but need to include also the transient part of the random force. We therefore analyse the projected equations with the random force from (7.2.9).

In Fig. 7.5(b) we observe that the KL divergence resulting from this more sophisticated approximation is close to zero, except initially – here the fact that we are not capturing the time-dependence of the transient part of the random force evidently matters. Beyond this early time regime, the values of the KL divergence are close to those we found previously when using the full time-dependent random force, as shown in Fig. 7.2. Put differently, the KL divergences from our approximation are indistinguishable from zero within the accuracy of our sampling. We show in the figure also the results for

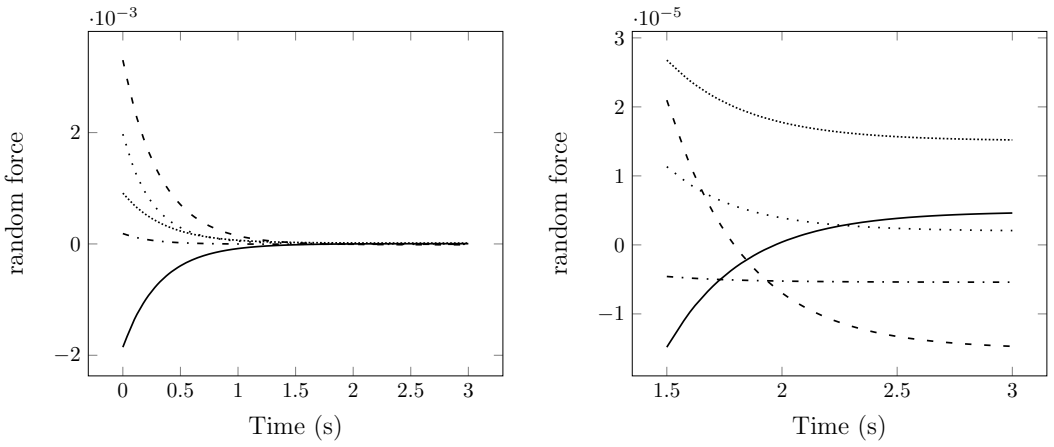


Figure 7.4: (a) The random force of δx_1 with a sample of different initial conditions. (b) There is a small but nonzero asymptote on each curve.

approximating the random force purely by its transient piece. These are essentially indistinguishable from the approximation that includes also the persistent contribution, in line with our earlier observation that in this example the persistent contribution is very small.

Returning to the naive approximation of approximating the random force only by its persistent piece, the reason for the large KL divergences that result can be seen by looking at the ratios of the variances in the predictions from the full reaction equations and approximated projected equations. Fig. 7.6 shows that there is a large difference in the variance of δx_1 taken from the reaction equations compared to the projected equations with constant random force and therefore the second term in equation (7.1.1) will be very large. Intuitively, the "persistent only" random force approximation leaves out the dominant source of noise and so drastically underestimates the variance. On the other hand, approximating using only the transient part, or the transient plus the persistent part, gives variance ratios that are close to unity except in the early time regime; see Fig. 7.6(b).

The above conclusions for this simple example system are confirmed when we look at the mean square error of the correlation matrices, which is sensitive also to correlation between fluctuations of different subnetwork concentrations. In Fig. 7.7 we see again

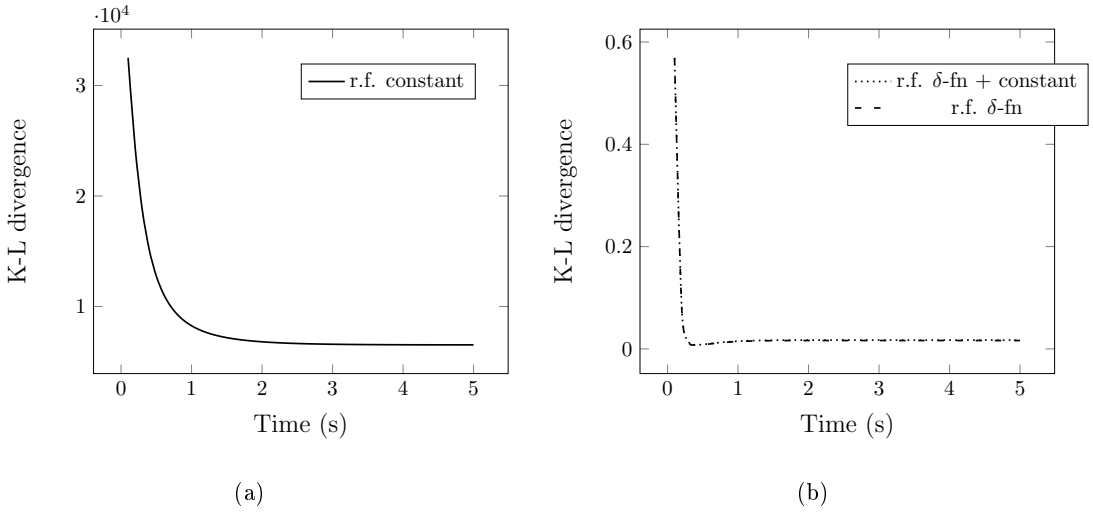


Figure 7.5: The KL divergence for δx_1 in the example model, for different choices of the random force. (a) The random force is approximated by a constant, i.e. only the persistent part is retained. (b) The random force is approximated by a δ-function (transient piece) and constant (persistent) piece, and only a δ-function; the results are indistinguishable visually.

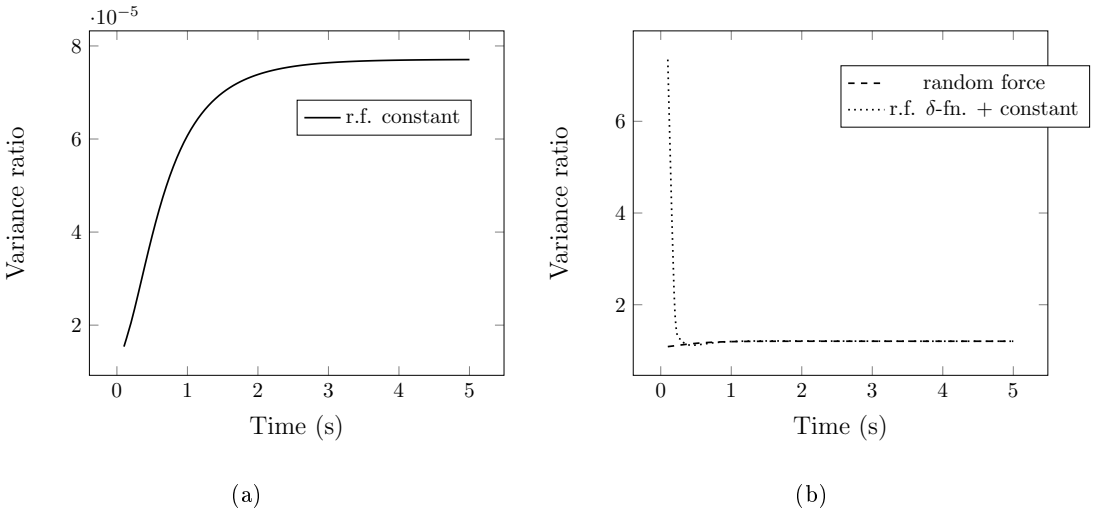


Figure 7.6: The ratio of the variances for δx_1 predicted by the full reaction equations and the projected equations with different choices for the random force. (a) Random force approximated using only the persistent piece. (b) Full random force, and approximation with a random force retaining both the transient and persistent piece.

that approximating the random force by a constant is not a good approximation of the random force: the mean square error is approximately one, which is indicative of the

approximate covariance matrices having much smaller entries than the true ones. On the other hand the error is small for the δ-function approximation with or without the constant piece (transient only and persistent plus transient, respectively), showing that both give good accuracy and that the persistent piece is not necessary to achieve this.

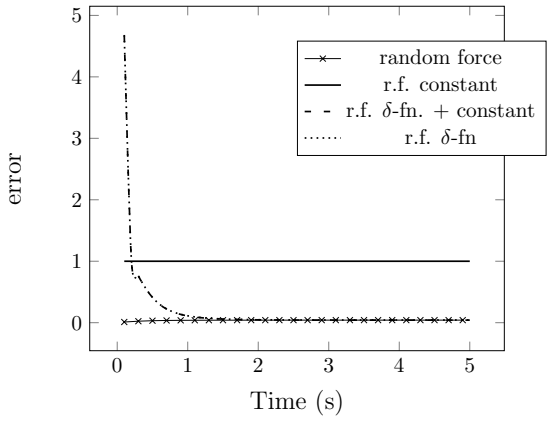


Figure 7.7: The mean square error of the correlation matrices of the example network for different approximations of the random force.

Next we consider the EGFR network to see how a larger system is affected by the presence of the random force resulting from random bulk initial conditions. Fig. 7.8 shows that the KL divergence for the projected equations with the full time-dependent random force, random force approximated by a δ-function plus constant part, and random force approximated by a δ-function are close to zero. For the first case this is expected as the linear projected equations should be exact; for the other two we observe again that approximation errors are significant only during an initial period of around 10s. By contrast one observes that approximating the random force as a constant, i.e. by its persistent part, does not at all give a good approximation as indicated by the extremely large KL values. As before one can check that these result from a serious underestimation of fluctuations. Looking at the mean square error of the correlation matrices we again see in Fig. 7.9 that there is a large difference in the correlation matrices of the reaction equations and projected equations with random force approximated as a constant. On the other hand, if we approximate the random force only by using the short time piece without the constant part $\boldsymbol{r}_\infty$, then we obtain results that are almost the same as when we include the constant part.

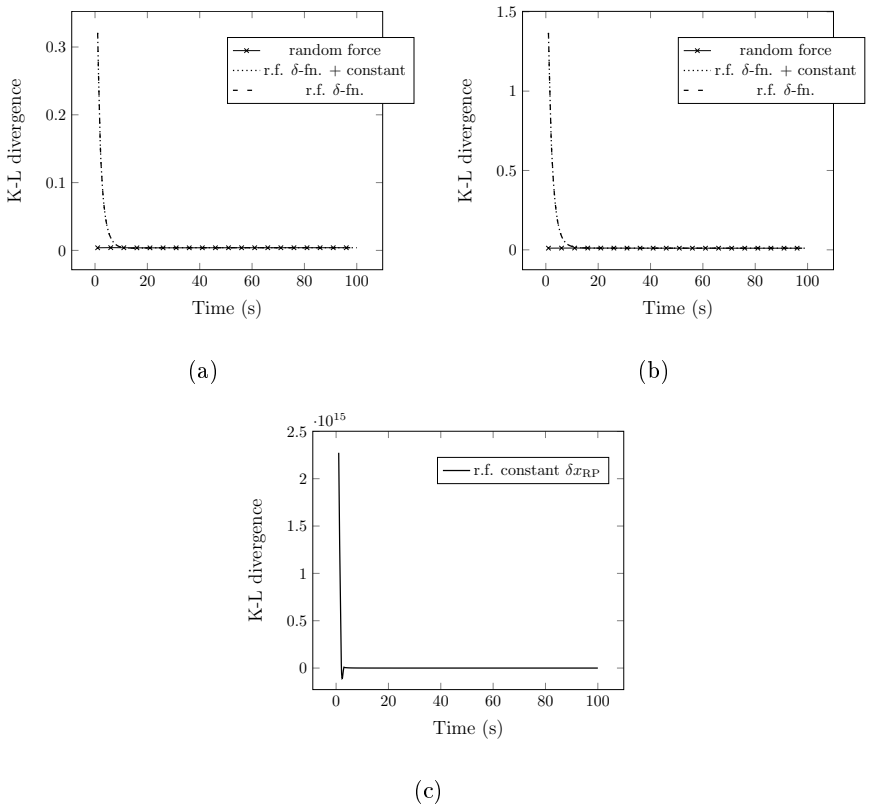


Figure 7.8: The KL divergence for proteins from the EGFR network for the full time-dependent linear random force and its various approximations. (a) Phosphorylated EGFR (RP); (b) Grb2; (c) Phosphorylated EGFR (RP).

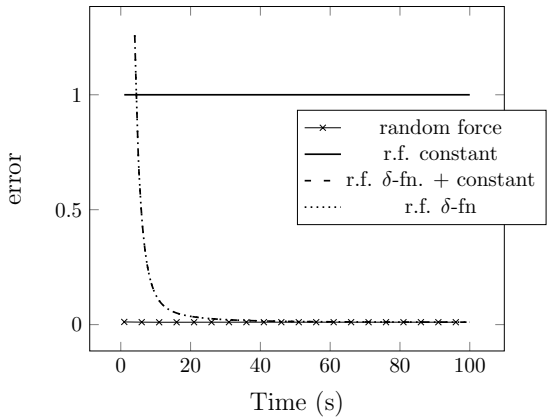


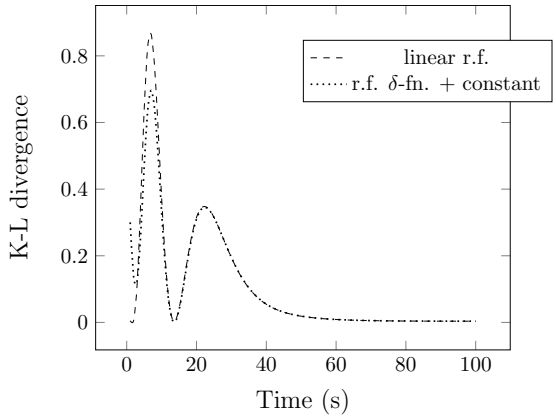
Figure 7.9: The mean square error of the correlation matrices for the EGFR subnetwork, for different approximations of the random force.

7.3 Nonlinear dynamics

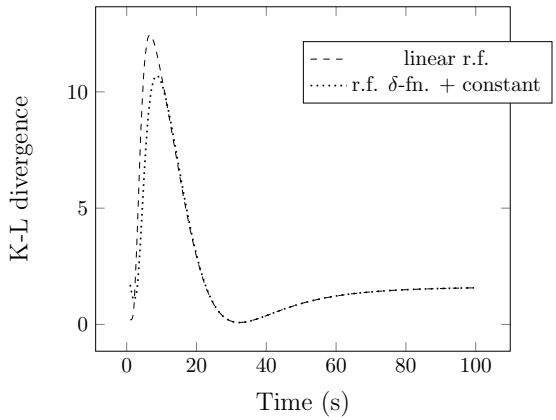
To model the nonlinear dynamics we also need to consider the best approximation of the random force. When we consider linearised dynamics the time evolution noise averages to zero, so the time evolution of the conditional averages that the projected equations track is by this source of noise. This is no longer the case with the nonlinear dynamics where in order to be able to calculate the memory functions in closed form we had to take the limit $\epsilon \to 0$. We will therefore assume that there is a separate parameter ϵ_0 setting the strength of fluctuations in the bulk initial conditions. This means we will sample bulk initial conditions with variance equal to $\langle \delta x^2 \rangle = \epsilon_0 y$. We sampled 100 bulk initial conditions and then calculated the KL divergences for different approximations of the random force as before.

In this initial exploration of random force effects in nonlinear dynamics, we consider the nonlinear projected equations with only the *linear* random force, i.e. not accounting for terms proportional to squares of bulk initial concentrations (see eq. (3.2.42)). We sampled bulk initial conditions with $\epsilon_0 = 0.001$ and the subnetwork initial conditions were chosen to maximise nonlinear effects subject to the constraint that the conserved quantities have the same value as at the steady state. Fig. 7.10 shows that this gives a good approximation to the reaction equations. This encourages us to try the approximation from (7.2.9) and we can see that even for the nonlinear dynamics we obtain reasonably accurate results when using this approximation. The quality of the approximation does differ depending on what concentration we focus on to assess fluctuations, with the fluctuations of phosphorylated EGFR (RP) being captured more accurately that those of Grb2. We would also expect that decreasing the size of ϵ_0 would give us results closer to the results of the linear dynamics.

An interesting feature in the KL divergence for phosphorylated EGFR (RP) is that it has a dip around $t = 15$s. To try and understand the reason for this we look at histograms at $t = 10$s, $t = 15$s and $t = 20$s to compare the results of the reaction equations and projected equations with the time-dependent linear random force, as shown in Fig. 7.11. We see that the histograms of the reaction equations and projected equations are different either side of the dip at $t = 10$s and $t = 20$s, but overlap more at the dip, accounting for the smaller KL divergence at this timestep.



(a)

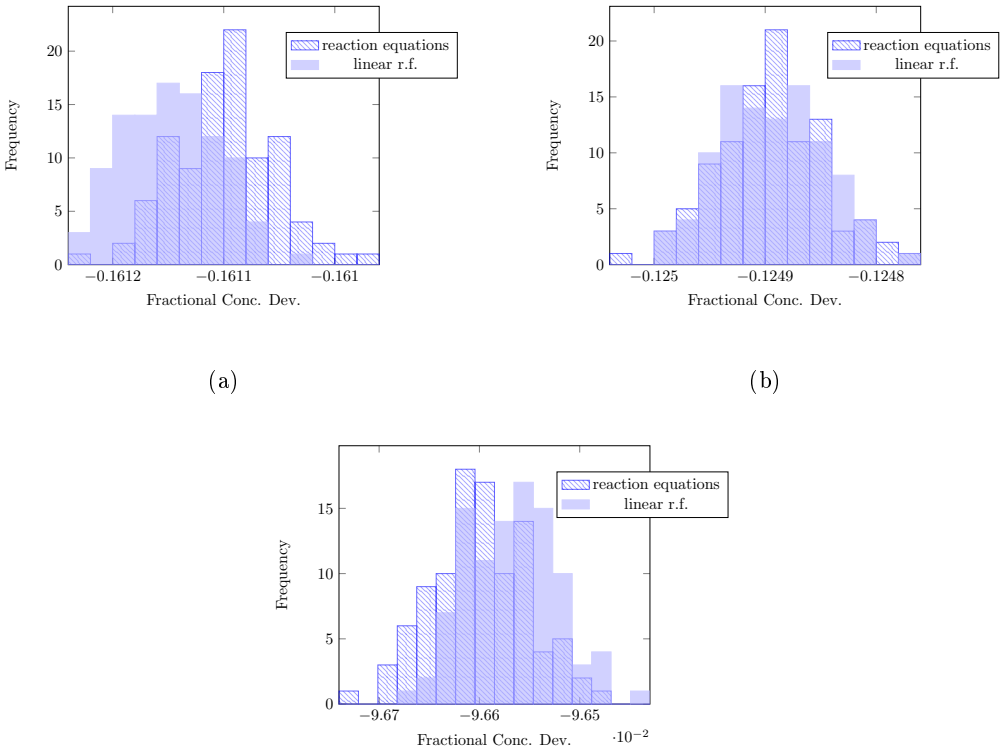


(b)

Figure 7.10: KL divergence between the predictions of the nonlinear reaction equations and projected equations, for the fluctuations of (a) phosphorylated EGFR (RP) and (b) Grb2. Shown are the results for the full time-dependent linear random force and its approximation as a sum of transient and a persistent piece.

7.4 Summary

We have analysed the random force contribution to the projected equations and have seen that it is possible to approximate the linearised random force by a δ-function and constant piece, which respectively capture the transient and persistent parts of the random force. The first contribution has the simple effect of perturbing the subnetwork initial conditions randomly. The combined "transient plus persistent" approximation gives accurate results for both the linearised and nonlinear dynamics, while the naive approximation, retaining only the persistent piece of the random force, is generally poor



(a) (b) (c)

Figure 7.11: Histograms of the reaction equations and projected equations with time-dependent linear random force, for phosphorylated EGFR (RP) at (a) $t = 10$s (b) $t = 15$s (c) $t = 20$s. At $t = 15$s the histograms show the most overlap.

because it drastically underestimates the strength of random force-induced fluctuations.

One obvious avenue for future work would to avoid the decoupling between ϵ and ϵ_0 in the fluctuation analysis for nonlinear dynamics. To avoid this somewhat artificial separation, one could go back to $\epsilon = \epsilon_0$ but solve the full nonlinear reaction equations including explicit stochasticity in the time evolution, averaging over multiple realisations to obtain true conditional averages of subnetwork concentrations for given subnetwork and (randomly sampled then fixed) bulk initial conditions. Distributions of these could then again be compared to the predictions from the projected equations, which would now be approximate, not only in their treatment of the random force, but also via the fact that the memory functions used are those for $\epsilon \to 0$ rather than the actual nonzero ϵ. Clearly the need to simulate stochastic nonlinear dynamics and to do so repeatedly

to obtain accurate conditional averages would add substantially to the computational effort for such numerical experiments, so we have not pursued them within the scope of this thesis.

An alternative approach would be to step away from the projection approach with its focus on conditional averages and switch to techniques that can track all sources of fluctuations including the stochasticity in the time evolution. Path integrals are a promising direction to consider here, but again lie outside the scope of this work.

Chapter 8

Conclusion

8.1 Summary

In this thesis, I have considered how to describe the dynamics of a subnetwork embedded in a bulk network. In Chapter 2, I looked at how to describe reaction equations for the dynamics of biochemical networks and introduced the projection method. I also summarised some different approaches to model reduction and subnetwork analysis.

Chapter 3 focused on applying the projected equations to protein interaction networks and how to calculate the memory functions. The projection approach allows one to obtain a set of dynamical equations for the concentration of molecular species in a chosen subnetwork that form part of a larger protein interaction network. These projected equations are closed, provided one neglects so-called random noise terms that represent one component of extrinsic noise, while the intrinsic noise can not be represented explicitly because the projection method only looks at evolution of conditional averages. For the linearised dynamics, the projection method gives results that are fully consistent with an explicit elimination of the bulk variables. Non-trivially, I was able to apply the projection method also to the full nonlinear dynamics, where in the limit of low copy number noise I found explicit formulas for the memory functions. These memory functions provide the weights, as a function of the time difference, with which past subnetwork states affect the present time evolution. I showed that they can be calculated from appropriate matrix representations of the dynamical (Fokker-Planck) and projec-

tion operators. In Sections 3.2.1 and 3.2.2 I analysed in some detail the properties of the linear and nonlinear memory functions, including their amplitudes and timescales. These provide insights into how the subnetwork interacts with the bulk, with e.g. negative memory amplitudes requiring the existence of ternary subnetwork-subnetwork-bulk complexes. Memory functions will also have contributions from species in the bulk that do not react directly with the subnetwork. The effects from these species will appear in the full time dependent memory function and could be analysed by looking at how plots of the memory function change when different bulk structures occur.

In Chapter 4, I applied the projection method to the EGFR network of Kholodenko et al. [13]. Here I illustrated the boundary structure of memory functions i.e. only species in the subnetwork that interact directly with the bulk have memory functions. I also showed how memory function amplitudes relate to subnetwork-bulk interaction structures. To understand memory function timescales, I used a channel decomposition. This is based on the fact that memory is generated by the past subnetwork state first affecting the bulk and then feeding back to the subnetwork at some later time. I showed that accordingly each memory function can be viewed as a sum of contributions from the different "source" and "receiver" channels in this feedback process. This allows one to identify which channels dominate the memory effects. The interpretation of each dominant channel could now be explored by designing experimentally tractable studies to interrogate parts of the subnetwork. I also gave a quantitative comparison of the accuracy of the projected equations versus simpler memoryless approximations. This showed that including nonlinear terms in the memoryless approximations does little to reduce approximation error, because the main error comes from neglecting linear memory terms. The nonlinear projected equations, on the other hand, were significantly more accurate than other approximations, by at least four orders of magnitude compared to the nearest memoryless competitor.

After analysing the projected equations I then looked in Chapter 5 at how the projection approach could be extended to systems involving enzymatic reactions represented as Michaelis-Menten terms. I started with a careful discussion of the conditions under which enzyme reactions represented in mass action form become equivalent to Michaelis-Menten kinetics. The result is a convenient scaling, of fast enzyme reaction rates and

simultaneously low enzyme concentrations, that exactly reproduces Michaelis-Menten equations in the limit where the relevant fast rate parameter γ grows large. Applying this construction, one can map a protein interaction network with Michaelis-Menten reactions to one with only unary and binary reactions; the main task is then to understand what limit is approached in the projected description when $\gamma \to \infty$. By analysing what fast enzyme degrees of freedom feature in the rate matrix and memory functions that define the projected equations, I showed that it is possible construct the projected equations directly in the large γ-limit, for both linearised and nonlinear dynamics, using a quasi-steady state elimination. This gives us effective unary reaction contributions to represent the Michaelis-Menten dynamics, with concentration-dependent reaction rates in the nonlinear case, and so allows one to constuct the projected equations without ever introducing enzymes explicitly. The resulting method significantly widens the range of biochemical reaction systems to which the projection approach can be applied; as one example I showed an application to a subnetwork of the EGFR reaction network of Kholodenko et al. [13].

In Chapter 6, I extended the projected equations to include perturbations by adding terms to the reaction equations that represent gene regulation. I first considered linear response where I showed that it is possible to split the effects of a bulk perturbation into the response from the perturbed bulk species onto the boundary and a perturbation of the boundary into the whole subnetwork. Both these terms can be split into transient and persistent contributions. If there is no flux across the boundary then I only obtained a transient perturbation and a persistent response; however in Section 6.2 I showed that a flux across the boundary will result in both a transient perturbation and a persistent response as well as a persistent perturbation and a transient response. This is illustrated with some small example networks. I also showed in Section 6.2.5.2 that adding gene regulation terms to the projected equations results in two timescales for the dynamics. A fast protein timescale and a slow gene regulation timescale. In Sec. 6.3 I looked at nonlinear response and I showed that this is more accurate for large perturbations, for example, gene knockdowns.

Finally, in Chapter 7, I analysed the effects of the random force when the bulk is not in steady state. I looked at different approximations of the random force and found that

one needs to keep the short time piece of the random force, which can be represented as a δ-function and an integral over the short time random force, to obtain an accurate result. This section links nicely with the perturbation calculation which helps with the understanding of why one needs both parts of the random force.

Projection operators have been applied to biochemical reaction networks by Thomas et al. [33]. Their method focuses on networks where there is timescale separation such that reactions in the bulk occur on a faster timescale than reactions in the subnetwork. The method developed in this thesis can be applied more generally to networks where there are no restrictions on the speed of reactions of the subnetwork and bulk. A benefit of the method of Thomas et al. [33] is that they explicitly track the noise and obtain a subnetwork with a new diffusion term that contains effects from both the subnetwork and the bulk. The projection approach shown in this thesis involves conditional averages and therefore we do not explicitly keep track of such dynamical noise. It would be possible to extend our approach to include intrinsic noise by integrating out the original bulk reaction equations directly. We would then have a set of subnetwork equations which contain a noise term with contributions from both the bulk and the subnetwork, $\tilde{\eta}(t) = \eta_s(t) + \int_0^t dt' \eta_b(t') e^{\boldsymbol{L}^{\mathrm{b,b}}(t-t')} \boldsymbol{L}^{\mathrm{b,s}}$. If we were to assume the bulk was fast we should then obtain the noise term from Thomas et al. [74].

8.2 Future work

The work I have presented in this thesis could be extended in a number of ways. The nonlinear subnetwork dynamics were obtained by including all possible products of subnetwork species (Section 3.2.2). This ensured that the random force was also zero to second order at zero time when we set the bulk initial conditions to zero. It would be interesting to investigate the tradeoff between computational efficiency and obtaining projected equations that are accurate due to small random force terms. If one includes all possible subnetwork products in the subnetwork then the projected equations are accurate but it is less efficient to calculate and solve them. Including in the subnetwork only the products that appear directly would make the memory function easier to calculate; however there would be components of the random force which would not automatically be small when the bulk is initially in steady state.

Throughout I have used the EGFR model from Kholodenko et al. [13] as my example network and selected a subnetwork and bulk; however there are many more known species that make up the EGFR network [49, 75]. It would then be possible to apply the projection method using a model with the network from Kholodenko et al. [13] as the selected subnetwork and the other known species as the bulk.

In the larger EGFR network not all rates are known, and even in the exisiting model of Kholodenko et al. [13] a number of rates are order of magnitude estimates. Using rate parameters sampled from a distribution as bulk reaction rates we could then look at the effects of a large unknown bulk on the subnetwork and at how much variation there is in memory functions when bulk rates are sampled [8]. This approach could also be used for more abstract models, for example, we could look at the limit of a very large (or infinite) bulk and how the distribution of the bulk reaction rates – as well as e.g. the connection structure of the bulk – affects the memory functions.

The memory function gives us information about how the subnetwork interacts with the bulk. In this thesis I have considered the forward problem of obtaining the projected equations and memory functions from the properties of the bulk. It would be useful to consider the inverse problem and use the memory function to try and learn something about the structure of the bulk. This could be achieved by first assuming the bulk has a simple form e.g. a tree. Then starting from this known structure of the bulk we can look at the form of the memory function. We can then see if it is possible to reverse the process so that we can find the structure of the bulk by looking at the memory function. In particular, we could try and identify the structure of the bulk that is directly connected to the boundary. We could then try and extend this to more complicated bulk structures, including loops. The inference of the structure of the bulk would first be done on model networks and then we may want to look at candidate bulk structures based on the structure of the EGFR network. This could be extended to models where we have experimental data. We would look at how to construct memory functions from data [76] and could then use this memory function to see what we can say about the structure of the bulk.

The projected equations have a boundary structure. If we have knowledge of a subnetwork that is embedded in a bulk network then the boundary structure could be used

as a starting point to identify which subnetwork species are on the boundary. If we assume that we have timecourse data for the subnetwork then we can work out the rate of change for each subnetwork species. If it is an interior species then we should be able to write this rate of change as a function of the other subnetwork species; however if it is a boundary species then this shouldn't work as we will be missing some information from its reactions with the bulk. This would enable us to tell which parts of the network are strongly connected to the bulk.

Chapter 6 shows the effects of perturbing the bulk. This work could be made more biologically applicable by comparing our results with experimental data taken, for example, from gene knockdown experiments. This would allow us to see the accuracy of the results from the nonlinear response.

The projection method with gene regulation could be extended by using more fine-grained models, for example, by using models that also include mRNA. We could then investigate how different separations of the subnetwork and bulk affect our results. For example, we could keep the protein interactions in the subnetwork and put the mRNA in the bulk. This could help us to see the best effective coarse graining of the model so that we know what information can be put in the bulk without losing too much knowledge of the system.

It would also be interesting to connect the theoretical results from this thesis with results from experiments. Firstly, we could look at what happens to the memory functions from the EGFR model if one of the nodes is removed. We could also look at additional examples of signalling pathways and how changes to rate constants affect the rest of these networks. These predictions could then be tested experimentally by using gene knockdowns to compare the predictions from the projected equations with the experimental results.

There have also been siRNA screens where a network is taken and specific nodes are knocked down one at a time. We could compare results from the projection approach with results from these siRNA screens. For example, Carlin et al. [77] look at the proteins which interact with Cdc42. We could then look at whether the projection approach would highlight the same Cdc42 interactors as being important.

We have analysed the random force statistics in Chapter 7; however this work could be

extended by doing more calculations to obtain a better understanding of the effects of the random force. We could look at using different divergence estimators that will allow more than just results for single species. This would give us a more detailed evaluation of the accuracy of different approximations of the random force. It has been shown [47] that extrinsic noise can have large effects on biochemical networks. Further analysis of the random force will also allow us to understand more about this component of the extrinsic noise.

It would be interesting to extend the projection method to also include effects of intrinsic noise. At least heuristically this could be done by adding noise back into the projected equations. For the linearised dynamics the projected equations could be calculated by integrating out the bulk stochastic reaction equations and substituting the solutions into the equations for the subnetwork species. The finite copy number noise then has an intrinsic part η_s and an extrinsic part η_b from the integrated out bulk - which is no longer white noise [33]. The extrinsic noise then has two contributions; the first from η_b and the second from the bulk initial conditions which is the random force. We could also look at including intrinsic noise into the equations for the nonlinear dynamics by only considering linear noise terms i.e. $\mathcal{O}(\epsilon\delta x)$. It would be interesting to see how the presence of intrisic noise affects the structure of the memory functions. We could also use path integrals to obtain an effective description of subnetwork dynamics that contains explicitly both the intrinsic and extrinsic noise.

Appendices

Appendix A

Protein species Table

EGF	Epidermal Growth Factor
R	Extracellular domain of the monomeric EGFR
R_a	EGF-EGFR complex
R_2	(EGF-EGFR)/(EGF-EGFR) dimer also known as R_a:R_a dimer
RP	Tyrosine phosphorylated EGFR
PLCγ	Phospholipase Cγ
Grb	Growth factor receptor-binding protein 2 (Grb2)
Shc	Src homology and collagen domain protein
SOS	Son of Sevenless homolog protein
E1	Enzyme for MM reaction between R_2 and RP
ER	Enzyme complex for MM reaction between R_2 and RP
E2	Enzyme for MM reaction between PLCγ and PLCγP
EP	Enzyme complex for MM reaction between PLCγ and PLCγP
E3	Enzyme for MM reaction between Shc and ShP
ES	Enzyme complex for MM reaction between Shc and ShP
RPL	RP-PLCγ
RPLP	RP-PLCγP/ Phosphorylated RPL
RG	RP-Grb2
RGS	RP-Grb2-SOS
RSh	RP-Shc
RShP	RP-ShP/ Phosphorylated RSh
RShG	RP-Shc-Grb2
RShGS	RP-Shc-Grb2-SOS
GS	Grb2-SOS
ShP	Phosphorylated Shc
ShG	ShP-Grb2
ShGS	Shc-Grb2-SOS
PLCγP	Phosphorylated PLCγ
PLCγPI	PLCγP translocated to membrane structures

Table A.1: Abbreviations used for EGFR network components following Kholodenko et al. [13], including a description of each enzyme added to the system to account for the Michaelis Menten (MM) reactions.

Appendix B

Projected equation solver

We can solve the projected equations by writing them as a set of differential equations. Let a_α be a set of subnetwork observables. Then the projected equations are written

$$\frac{\partial}{\partial t}a_\alpha = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \int_0^t dt' \sum_\beta a_\beta(t')M_{\beta\alpha}(t-t') + r_\alpha(t) \tag{B.0.1}$$

where the memory function can be written as

$$M_{\beta\alpha}(t-t') = \sum_a m^a_{\beta\alpha} e^{-\lambda_a(t-t')} \tag{B.0.2}$$

for $\lambda_a > 0$. Substituting (B.0.2) into the projected equations we obtain

$$\frac{\partial}{\partial t}a_\alpha = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \int_0^t dt' \sum_{\beta,a} a_\beta(t')m^a_{\beta\alpha} e^{-\lambda_a(t-t')} + r_\alpha(t) \tag{B.0.3}$$

We then define

$$c^a_\alpha(t) = \int_0^t dt' \sum_\beta a_\beta(t')m^a_{\beta\alpha} e^{-\lambda_a(t-t')} \tag{B.0.4}$$

where we can write equations for the $c^a_\alpha(t)$ as

$$\frac{\partial}{\partial t}c^a_\alpha = -\lambda_a c^a_\alpha + \sum_\beta a_\beta m^a_{\beta\alpha} \tag{B.0.5}$$

The set of equations we need to solve to obtain a solution for the projected equations are then

$$\frac{\partial}{\partial t}a_\alpha = \sum_\beta a_\beta\Omega_{\beta\alpha} + \sum_a c^a_\alpha + r_\alpha(t) \tag{B.0.6}$$

$$\frac{\partial}{\partial t}c^a_\alpha = -\lambda_a c^a_\alpha + \sum_\beta a_\beta m^a_{\beta\alpha} \tag{B.0.7}$$

with initial conditions $c^a_\alpha(0) = 0$.

B.1 Projected equations with perturbations

We write the perturbed projected equations

$$\frac{\partial}{\partial t}a_\alpha = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \int_0^t dt' \sum_\beta a_\beta(t')M_{\beta\alpha}(t-t') + \int_0^t dt' \sum_\beta h_\beta R_{\beta\alpha}(t-t') + r_\alpha(t) \tag{B.1.1}$$

We must therefore also include the response function when solving the equations. The nonlinear projected equations have extra contributions to the memory of $\mathcal{O}(h)$. We can then write the memory in two parts where

$$\sum_a m^a_{\beta\alpha} e^{-\lambda_a(t-t')} \tag{B.1.2}$$

contains the $\mathcal{O}(\delta x)$ and $\mathcal{O}(\delta x^2)$ memory contribution as well as the $\mathcal{O}(h\delta x)$ terms from (6.3.3). We write the $\mathcal{O}(h\delta x)$ terms from (6.3.4) as

$$\sum_a m'^a_{\beta\alpha} \cdot (t-t') e^{-\lambda_a(t-t')} \tag{B.1.3}$$

and the response function is written

$$\sum_\beta h_\beta R_{\beta\alpha} = \sum_\beta h_\beta n^a_{\beta\alpha} e^{-\lambda_a(t-t')} \tag{B.1.4}$$

We can then write the perturbed projected equations as

$$\frac{\partial}{\partial t}a_\alpha = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \sum_a \int_0^t dt' \sum_\beta a_\beta(t') \left(m^a_{\beta\alpha} + m'^a_{\beta\alpha} \cdot (t-t') + h_\beta n^a_{\beta\alpha}\right) e^{-\lambda_a(t-t')} + r_\alpha(t) \tag{B.1.5}$$

Defining

$$c^a_\alpha = \sum_a \int_0^t dt' \sum_\beta a_\beta(t') \left(m^a_{\beta\alpha} + m'^a_{\beta\alpha} \cdot (t-t') + h_\beta n^a_{\beta\alpha}\right) e^{-\lambda_a(t-t')} \tag{B.1.6}$$

the projected equations can be written

$$\frac{\partial}{\partial t}a_\alpha = \sum_\beta a_\beta(t)\Omega_{\beta\alpha} + \sum_\alpha c^a_\alpha + r_\alpha(t) \tag{B.1.7}$$

The time derivative of the c^a_α is

$$\frac{\partial}{\partial t}c^a_\alpha = -\lambda_a c^a_\alpha + \sum_\beta a_\beta m^a_{\beta\alpha} + \sum_\beta h_\beta n^a_{\beta\alpha} + d^a_\alpha \tag{B.1.8}$$

where we have defined

$$d_\alpha^a = \int_0^t dt' \sum_\beta a_\beta(t') m'^a_{\beta\alpha} e^{-\lambda_a (t-t')} \tag{B.1.9}$$

and we can then write

$$\frac{\partial}{\partial t} d_\alpha^a = -\lambda_a d_\alpha^a + \sum_\beta a_\beta m'^a_{\beta\alpha} \tag{B.1.10}$$

To solve the projected equations we must solve equations (B.1.7), (B.1.8) and (B.1.10) with initial conditions $c_\alpha^a(0) = d_\alpha^a(0) = 0$.

Appendix C

Oscillations

Protein interaction networks don't always have a unique steady state, but can also exhibit bistability or oscillations. Negative feedback and ultasensitivity or effects from multisite phosphorylation can cause oscillations in mitogen-activated protein kinase cascades [78, 79]. To look at what happens with the projection approach when there is an oscillating system first consider the minimal mass-action oscillating system [80]

$$\begin{aligned} S + X &\xrightarrow{k_1} 2X \\ X + Y &\xrightarrow{k_2} 2Y \\ Y &\xrightarrow{k_3} P \end{aligned} \tag{C.0.1}$$

where S and P denote constant substrates and products [81]. This gives the differential equations

$$\begin{aligned} \partial_t X &= k_1 S X - k_2 X Y \\ \partial_t Y &= k_2 X Y - k_3 Y \end{aligned} \tag{C.0.2}$$

and solving these equations gives oscillatory solutions for X and Y as shown in Fig. C.1.

This system can be converted into a standard system with binary reactions if we assume that Y becomes S again, where S is now a non-constant reactant.

$$\begin{aligned} S + X &\xrightarrow{k_1} 2X \\ X + Y &\xrightarrow{k_2} 2Y \\ Y &\xrightarrow{k_3} S \end{aligned} \tag{C.0.3}$$

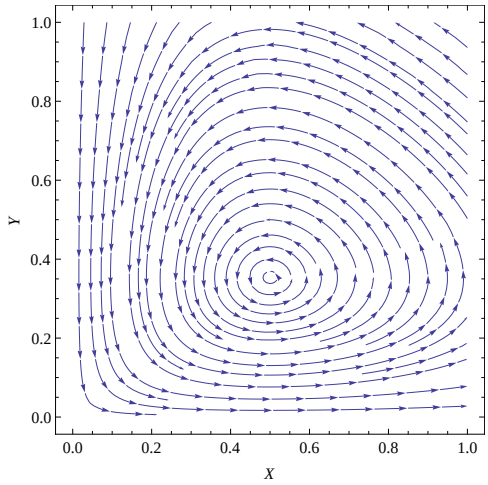


Figure C.1: The phase portrait of the original system (C.0.1).

These equations have the conservation law $X + Y + S = C$, where C is the constant total concentration. If the total concentration is large then (C.0.3) looks like the original system (C.0.1). This is because taking the original system of equations (C.0.4) and replacing S with the conservation law we obtain

$$\begin{aligned} \partial_t X &= k_1(C - X - Y)X - k_2 XY \\ \partial_t Y &= k_2 XY - k_3 Y \end{aligned} \tag{C.0.4}$$

When C is large such that $k_1 C = \tilde{k}_1$ then we find that $k_1(C - X - Y) \to \tilde{k}_1$ because X and Y are bounded as it is an oscillating system.

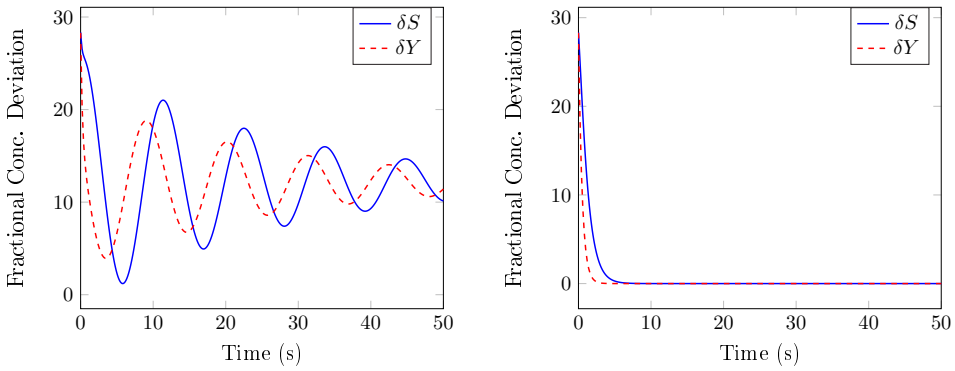
To convert the system to fit in with a binary and unary reaction system complexes can be added and the system then becomes

$$\begin{aligned} X + S &\underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} XS \\ XS &\underset{k_2^-}{\overset{k_2^+}{\rightleftharpoons}} 2X \\ X + Y &\underset{k_3^-}{\overset{k_3^+}{\rightleftharpoons}} XY \\ XY &\underset{k_4^-}{\overset{k_4^+}{\rightleftharpoons}} 2Y \\ Y &\underset{k_5^-}{\overset{k_5^+}{\rightleftharpoons}} S \end{aligned} \tag{C.0.5}$$

The fact that the original system oscillates suggests that there should also be parameter choices in this extended system where we obtain oscillatory relaxation. For a system

k_1^+	2	k_1^-	1
k_2^+	1	k_2^-	0.001
k_3^+	2	k_3^-	0.002
k_4^+	1	k_4^-	0.001
k_5^+	1	k_5^-	0.001

Table C.1: The reaction rates needed to make system (C.0.5) oscillate. One could take the unit of concentration to be nMol and of time to be seconds.



(a) (b)

Figure C.2: The solutions to (a) the projected equations; (b) the projected equations without the memory function.

to oscillate the drift matrix $\boldsymbol{A}$ from (3.1.13) must have imaginary eigenvalues. A set of reaction rates that will cause oscillatory relaxation in the system are shown in Table C.1. In this example the steady state value of X is unconstrained, setting this to be 0.5 to match the original system ensures that the equations for the fluctuations about the mean steady state oscillate around a stable fixed point.

Applying the projection method to this system where the subnetwork is chosen to be $(\delta S, \delta Y)$ and looking at the linearised dynamics we find that the equations containing only the rate matrix but no memory function don't oscillate, but require the memory function to obtain the oscillations, as shown in Fig. C.2. This shows the importance of the memory function if we want to obtain an accurate description of the dynamics.

References

[1] U. S. Bhalla. Understanding complex signaling networks through models and metaphors. *Prog. Biophys. Mol. Bio.*, 81:45–65, 2003.

[2] J. Ackermann, J. Einloft, J. Nöthen, and I. Koch. Reduction techniques for network validation in systems biology. *J. Theor. Biol.*, 315:71–80, 2012.

[3] C. Conradi, D. Flockerzi, J. Raisch, and J. Stelling. Subnetwork analysis reveals dynamic features of complex (bio)chemical networks. *Proc. Natl. Acad. Sci. U. S. A.*, 104(49):19175–19180, 2007.

[4] A. Shojaie and G. Michailidis. Penalized principal component regression on graphs for analysis of subnetworks. In *Adv. Neur. In.*, pages 2155–2163. 2010. editor: J. Lafferty and C. K. I. Williams and J. Shawe-Taylor and R.S. Zemel and A. Culotta.

[5] M. S. Okino and M. L. Mavrovouniotis. Simplification of mathematical models of chemical reaction systems. *Chem. Rev.*, 98(2):391–408, 1998.

[6] O. Radulescu, A. N. Gorban, A. Zinovyev, and V. Noel. Reduction of dynamical biochemical reactions networks in computational biology. *Front. Genet.*, 3(131): 131, 2012. ISSN 1664-8021.

[7] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

[8] T. R. Xu, V. Vyshemirsky, A. Gormand, A. von Kriegsheim, M. Girolami, G. S. Baillie, D. Ketley, A. J. Dunlop, G. Milligan, M. D. Houslay, and W. Kolch. Inferring signaling pathway topologies from multiple perturbation measurements of specific biochemical species. *Science Signaling*, 3(113), 2010.

[9] P. S. Swain, M. B. Elowitz, and E. D. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc. Natl. Acad. Sci. U. S. A.*, 99(20):12795–12800, 2002.

[10] J. Paulsson. Summing up the noise in gene networks. *Nature*, 427(6973):415–418, 2004.

[11] V. Shahrezaei and P. S. Swain. The stochastic nature of biochemical networks. *Curr. Opin. Biotechnol.*, 19(4):369–374, 2008.

[12] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

[13] B. N. Kholodenko, O. V. Demin, G. Moehren, and J. B. Hoek. Quantification of short term signaling by the epidermal growth factor receptor. *The J. of Bio. Chem.*, 274(42):30169–30181, 1999.

[14] G. Karlebach and R. Shamir. Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.*, 9(10):770–780, 2008. ISSN 1471-0072.

[15] D. J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nat. Rev. Genet.*, 10(2):122–133, 2009.

[16] B. Kholodenko, M. B. Yaffe, and W. Kolch. Computational approaches for analyzing information flow in biological networks. *Science Signaling*, 5(220):re1–re1, 2012.

[17] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry*. Elseiver, London, 3rd edition, 2007.

[18] D. T. Gillespie. The chemical Langevin equation. *J. Chem. Phys.*, 113(1):297–306, 2000.

[19] C. W. Gardiner. *Stochastic methods: a handbook for the natural and social sciences*. Springer Series in Synergetics. Springer-Verlag, Berlin, 4th edition edition, 2009.

[20] J. Elf and M. Ehrenberg. Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Res*, 13:2475–2484, 2003.

[21] R. Grima, P. Thomas, and A. V. Straube. How accurate are the nonlinear chemical Fokker-Planck and chemical Langevin equations? *J. Chem. Phys.*, 135(8):084103, 2011.

[22] M. Doi. Stochastic theory of diffusion-controlled reaction. *J. Phys. A*, 9(9):1479–1495, 1976.

[23] L. Peliti. Path integral approach to birth-death processes on a lattice. *Journal De Physique*, 46(9):1469–1483, 1985.

[24] T. Brett and T. Galla. Stochastic processes with distributed delays: Chemical langevin equation and linear-noise approximation. *Phys. Rev. Lett.*, 110:250601, 2013.

[25] P. Thomas, C. Fleck, R. Grima, and N. Popovic. System size expansion using feynman rules and diagrams. *J. Phys. A: Math. Gen.*, 2014.

[26] B. Bravi and P. Sollich. Gaussian variational approximation in biochemical networks. *In preparation*, 2014.

[27] H. Mori. Transport collective motion and brownian motion. *Prog. Theor. Phys.*, 33 (3):423, 1965.

[28] J. P. Hansen and I. R. McDonald. *Theory of Simple Liquids.* Academic Press, San Diego, 2nd edition, 1990.

[29] F. Ritort and P. Sollich. Glassy dynamics of kinetically constrained models. *Adv. Phys.*, 52(4):219–342, 2003.

[30] W. Götze and L. Sjögren. The mode coupling theory of structural relaxations. *Transp. Theory Stat. Phys.*, 24(6-8):801–853, 1995.

[31] Z. J. Huang, Y. Chu, and J. Hahn. Model simplification procedure for signal transduction pathway models: An application to il-6 signaling. *Chem. Eng. Sci.*, 65(6): 1964–1975, 2010.

[32] C. W. Gardiner. Adiabatic elimination in stochastic-systems .1. Formulation of methods and application to few-variable systems. *Phys. Rev. A*, 29(5):2814–2822, 1984.

[33] P. Thomas, R. Grima, and A. V. Straube. Rigorous elimination of fast stochastic variables from the linear noise approximation using projection operators. *Phys. Rev. E*, 86(4):041110, 2012.

[34] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761):C47–C52, 1999.

[35] H. Conzelmann, J. Saez-Rodriguez, T. Sauter, E. Bullinger, F. Allgower, and E. D. Gilles. Reduction of mathematical models of signal transduction networks: simulation-based approach applied to egf receptor signalling. *IEE Systems Biology*, 1(1):159–69, 2004.

[36] J. Saez-Rodriguez, A. Kremling, and E. D. Gilles. Dissecting the puzzle of life: modularization of signal transduction networks. *Computers & Chemical Engineering*, 29(3):619–629, 2005.

[37] M. Sunnaker, G. Cedersund, and M. Jirstrand. A method for zooming of nonlinear models of biochemical systems. *BMC Syst. Biol.*, 5:140, 2011.

[38] W. Liebermeister, U. Baur, and E. Klipp. Biochemical network models simplified by balanced truncation. *FEBS J.*, 272(16):4034–4043, 2005.

[39] M. Apri, M. de Gee, and J. Molenaar. Complexity reduction preserving dynamical behavior of biochemical networks. *J. Theor. Biol.*, 304:16 – 26, 2012.

[40] A. C. C. Coolen and S. Rabello. Generating functional analysis of complex formation and dissociation in large protein interaction networks. *International Workshop On Statistical-mechanical Informatics 2009 (iw-smi 2009)*, 197:012006, 2009.

[41] W. Götze. Aspects of structural glass transitions. In J. P. Hansen, D. Levesque, and J. Zinn-Justin, editors, *Liquids, freezing and glass transition*, pages 287–503, Amsterdam, 1991. North-Holland.

[42] M. Beck, A. Schmidt, J. Malmstroem, M. Claassen, A. Ori, A. Szymborska, F. Herzog, O. Rinner, J. Ellenberg, and R. Aebersold. The quantitative proteome of a human cell line. *Mol Syst Biol*, 7:549, 2011.

[43] N. Nagaraj, J. R. Wisniewski, T. Geiger, J. Cox, M. Kircher, J. Kelso, S. Paeaebo, and M. Mann. Deep proteome and transcriptome mapping of a human cancer cell line. *Mol Syst Biol*, 7:548, 2011.

[44] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, H. Lehrach, and R. Herwig. *Systems Biology: A Textbook*. Wiley-VCH, Weinheim, 2009.

[45] N. G. van Kampen. The equilibrium distribution of a chemical mixture. *Physics Letters*, 59A:333–334, 1976.

[46] G. C. Wick. The evaluation of the collision matrix. *Phys. Rev.*, 80:268–272, 1950.

[47] V. Shahrezaei, J. F. Ollivier, and P. S. Swain. Colored extrinsic fluctuations and stochastic gene expression. *Mol Syst Biol*, 4:196, 2008.

[48] H. S. Wiley, S. Y. Shvartsman, and D. A. Lauffenburger. Computational modeling of the egf-receptor system: a paradigm for systems biology. *Trends In Cell Biology*, 13(1):43–50, 2003.

[49] K. Oda, Y. Matsuoka, A. Funahashi, and H. Kitano. A comprehensive pathway map of epidermal growth factor receptor signaling. *Mol Syst Biol*, 1(2005.0010): 1–9, 2005.

[50] N. Normanno, A. D. Luca, C. Bianco, L. Strizzi, M. Mancino, M. R. Maiello, A. Carotenuto, G. D. Feo, F. Caponigro, and D. S. Salomon. Epidermal growth factor receptor (egfr) signaling in cancer. *Gene*, 366(1):2 – 16, 2006. ISSN 0378-1119.

[51] J. Murray. *Mathematical Biology I. An Introduction*. Springer, New York, 2001.

[52] J. T. Day. Note on numerical solution of integro-differential equations. *Comput. J.*, 9(4):394, 1967.

[53] E. Hairer and G. Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems*. Springer Berlin Heidelberg, 2nd rev. edition, 1996. ISBN 9783642052224.

[54] V. Henri. Théorie générale de l'action de quelques diastases. *C. R. Acad. Sci. Paris*, 135:916–919, 1902.

[55] L. Michaelis and M. L. Menten. Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333, 1913.

[56] J. Keener and J. Sneyd. *Mathematical Physiology I: Cellular Physiology*. Springer, 2nd edition, 2009.

[57] G. E. Briggs and J. B. S. Haldane. A note on the kinetics of enzyme action. *Biochem. J.*, 19(2):338–339, 1925.

[58] H. M. Sauro. *Enzyme Kinetics for Systems Biology*. Ambrosius Publishing, 2nd edition, 2013.

[59] L. Segel and M. Slemrod. The quasi-steady-state assumption: a case study in perturbation. *SIAM Review*, 31:446–477, 1989.

[60] D. D. Van Slyke and G. E. Cullen. The mode of action of urease and of enzymes in general. *J. Biol. Chem.*, 19(2):141–180, 1914.

[61] J. Berg, J. Tymoczko, and L. Stryer. *Biochemistry*. New York, 5th edition, 2002.

[62] K. J. Rubin, K. Lawler, P. Sollich, and T. Ng. Memory effects in biochemical networks as the natural counterpart of extrinsic noise. *J. Theor. Biol.*, 357:245–267, 2014.

[63] M. A. Woodbury. *Inverting modified matrices*. Statistical Research Group, Memo. Rep. no. 42. Princeton University, 1950.

[64] B. N. Kholodenko, J. B. Hoek, H. V. Westerhoff, and G. C. Brown. Quantification of information transfer via cellular signal transduction pathways. *FEBS Lett.*, 414 (2):430–434, 1997.

[65] F. J. Bruggeman, H. V. Westerhoff, J. B. Hoek, and B. N. Kholodenko. Modular response analysis of cellular regulatory networks. *J. Theor. Biol.*, 218(4):507–520, 2002.

[66] N. Shayeghi, T. Ng, and A. C. C. Coolen. Direct response analysis in cellular signalling networks. *J. Theor. Biol.*, 304:219–225, 2012. ISSN 0022-5193.

[67] A. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, 149 (4):1633–1648, 1998.

[68] P. S. Swain. Efficient attenuation of stochasticity in gene expression through post-transcriptional control. *J. Mol. Biol.*, 344(4):965–976, 2004.

[69] M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biol.*, 7(3):R25, 2006.

[70] E. M. Ozbudak, M. Thattai, I. Kurtser, A. D. Grossman, and A. van Oudenaarden. Regulation of noise in the expression of a single gene. *Nat. Genet.*, 31(1):69–73, 2002.

[71] J. Berg. Out-of-equilibrium dynamics of gene expression and the jarzynski equality. *Phys. Rev. Lett.*, 100(18):188101, 2008.

[72] M. Fuchs and M. E. Cates. Integration through transients for brownian particles under steady shear. *J. Phys.-Condens. Mat.*, 17(20):S1681–S1696, 2005.

[73] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

[74] P. Thomas, A. V. Straube, and R. Grima. Communication: Limitations of the stochastic quasi-steady-state approximation in open biochemical reaction networks. *J. Chem. Phys.*, 135(18):181103, 2011.

[75] A. Kiyatkin, E. Aksamitiene, N. I. Markevich, N. M. Borisov, J. B. Hoek, and B. N. Kholodenko. Scaffolding protein grb2-associated binder 1 sustains epidermal growth factor-induced mitogenic and survival signaling by multiple positive feedback loops. *J. Biol. Chem.*, 281(29):19925–19938, 2006.

[76] M. Uranagase and T. Munakata. Generalized Langevin equation revisited: mechanical random force and self-consistent structure. *J. Phys. A*, 43(45):455003, 2010.

[77] L. M. Carlin, R. Evans, H. Milewicz, L. Fernandes, D. R. Matthews, M. Perani, J. Levitt, M. D. Keppler, J. Monypenny, T. Coolen, P. R. Barber, B. Vojnovic,

K. Suhling, F. Fraternali, S. Ameer-Beg, P. J. Parker, N. S. B. Thomas, and T. Ng. A targeted sirna screen identifies regulators of cdc42 activity at the natural killer cell immunological synapse. *Science Signaling*, 4(201):ra81, 2011.

[78] B. N. Kholodenko. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *Eur. J. Biochem.*, 267(6): 1583–1588, 2000.

[79] V. Chickarmane, B. N. Kholodenko, and H. M. Sauro. Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation. *J. Theor. Biol.*, 244(1): 68–76, 2007.

[80] A. J. Lotka. Undamped oscillations derived from the law of mass action. *J. Am. Chem. Soc.*, 42:1595–1599, 1920.

[81] T. Wilhelm. The smallest chemical reaction system with bistability. *BMC Syst. Biol.*, 3:90, 2009.